DR. D. Y. PATIL SCHOOL OF SCIENCE & TECHNOLOGY
DR. D. Y. PATIL VIDYAPEETH, PUNE
(Deemed to be University)
(Accredited (3rd cycle) by NAAC with a CGPA of 3.64 on four-point scale at 'A++' Grade)
(Declared as Category - I University by UGC Under Graded Autonomy Regulations, 2018)
(An ISO 9001: 2015 and 14001:2015 Certified University and Green Education Campus)

BCA detailed syllabus for Semester III to VIII

Dr. D. Y. Patil Vidyapeeth, Dr. D. Y. Patil School of science & Technology
Second Year BCA (2023 Course)
(With effect from Academic Year 2023-24)

SEMESTER III

Course Code	Course Type	Course Name	Teaching Scheme			Examination Assessment Scheme				Credit scheme			
			Lecture	Tutorial	Practical	CA	End Sem	Practical	Total	L	T	P	C
BCA-CA-301	Major	Data structures	3	0	4	40	60	100	200	3	0	4	5
BCA-CA-302	Major	Advanced Database Management System	4	0	0	40	60	-	100	4	0	0	4
BCA-CA-303	Major	Operating System Principles	3	0	4	40	60	100	200	3	0	4	5
PCC-CA-301	VA	Project Management	1	0	2	20	30	-	50	1	0	2	2
PEC-CA-301	DSE	Discipline Specific-3	2	0	4	40	60	100	200	2	0	4	4
HSMC-CA-301	AEC	Ability/Skill Enhancement	2	0	0	50	-	-	50	2	0	0	2
			15	**0**	**14**	**230**	**270**	**300**	**800**	**15**	**0**	**14**	**22**

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year BSc (2024-25 Course)

BSC-CA-301 : Data Structures

Teaching Scheme:	**Credit**	**Examination Scheme:**
TH: 3Hours/Week	**5**	**Internal (TH): 40 Marks** **External (TH): 60 Marks**

Prerequisite Courses, if any:

- Problem Solving and Programming In C

Course Objectives:

- To understand the basic concepts in data structure.
- To discuss various algorithmic strategies to solve real life problems.
- To acquaint the learner various data searching and sorting techniques.
- To identify and use the appropriate data structure for various real life problems using computer languages.
- To understand the concepts of linear, non-linear data structures with its complexities.
- To understand and efficiently apply various data structures

Course Outcomes:

On completion of the course, learner will be able to–

CO1: To understand the need of data structures.
CO2: To learn to apply the algorithm complexity techniques for various estimations.
CO3: To use organized data structure to solve various problem statements.
CO4: To develop the solutions to social issues using NP Complete theory.
CO5: To distinguish the use of various structures in solving problems.
CO6: To understand the usage of appropriate data structures to implement algorithms.

Course Contents

Unit I	**Introduction to Data Structures**	**(06 Hours)**

Introduction, Need of Data Structure, Fundamental Concepts: Data and information, Data type, Abstract Data Type, Types of Data Structures, Algorithms: Problem Solving, Introduction to algorithm, Characteristics of algorithm, Algorithm design tools: Pseudo-code and flowchart Complexity of algorithm: Space complexity, Time complexity, Asymptotic notation- Big-O, Theta and Omega, Finding complexity using step count method.

#Exemplar/Case Studies	Problems on time complexity calculation.
Mapping of Course Outcomes for Unit I	CO1

Unit II	Array	(07 Hours)
Overview of Array, Array as an Abstract Data Type, Operations on Array, Multidimensional Arrays: Two-dimensional arrays, n-dimensional arrays, Storage Representation and their Address Calculation: Row major and Column Major Array applications – Searching: Sequential search, Sentinel search, Binary Search, Fibonacci Search Sorting: Internal, External, Stable, In-place Sorting, Sorting Methods- Bubble Sort, Insertion Sort, Selection Sort, Merge Sort, Quick Sort, Bucket Sort.		
#Exemplar/Case Studies	Comparison of searching & sorting methods in terms of complexity.	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**Linked Lists**	**(07 Hours)**
Linked List as an ADT, Dynamic implementation of Linked List, Types of Linked List – Singly, Doubly, Circular, Operations on Linked List - create, traverse, insert, delete, search, sort, reverse, concatenate, merge, time complexity of operations. Applications of Linked List – Polynomial representation, Addition of two polynomials Generalized linked list – concept, representation, multiple-variable polynomial representation using generalized list.		
#Exemplar/Case Studies	Study and analyze use of linked lists in Operating Systems.	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**Stacks & Queues**	**(08 Hours)**
Stack: Concept of Stack, Stack as an ADT, Stack Implementation using sequential and linked organization, Stack Operations Applications of Stack: Recursion, converting expressions from infix to postfix or prefix form, evaluating postfix or prefix form. Queue: Concept of Queues, Queues as an ADT, Implementation of queue using array and linked organization, Queue Operations, Types of Queue- circular queue, double ended queue, priority queue Applications of Queue– CPU Scheduling in multiprogramming environment, Round robin algorithm		
#Exemplar/Case Studies	Study and analyze use of Priority queue in bandwidth management	

Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Trees**	**(07 Hours)**
Tree : Trees and binary trees-concept and terminology, Expression tree, Binary tree as an ADT Binary search tree: Binary search tree as an ADT(Insert Search Delete, level wise Display), Recursive and Non recursive algorithms for binary search tree traversals Threaded binary tree: Concept of threaded binary tree. Preorder and In-order traversals of in-order threaded binary tree Applications of trees.		
#Exemplar/Case Studies	Use of binary tree in expression tree-evaluation and Huffman's coding	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**Graphs**	**(06 Hours)**
Graph -Concept and terminologies, Graph as an ADT, Representation of graphs using adjacency matrix and adjacency list, Breadth First Search traversal, Depth First Search traversal, Prim's and Kruskal's algorithms for minimum spanning tree, Shortest path using Dijkstra's algorithm, All pairs shortest paths- Flyod-Warshall Algorithm, topological sorting.		
#Exemplar/Case Studies	Study and analyze working of Google map	
Mapping of Course Outcomes for Unit VI	CO6	
Learning Resources		

Text Books:

1. Horowitz, Sahani, Dinesh Mehata, ―Fundamentals of Data Structures in C++‖, Galgotia Publisher, ISBN: 8175152788, 9788175152786.
2. Y. Langsam, M. Augenstin, A. Tannenbaum, "Data Structures using C and C++", 2nd Edition, Prentice Hall of India, 2002, ISBN-81-203-1177-9.

Reference Books:

1. Classic Data Structures-D. Samanta, Prentice Hall India Pvt. Ltd.
2. Data Structures using C and C++- Yedidyah Langsam, Moshe J. Augenstein, Aaron M. Tenenbaum, Pearson Education
3. Data Structures: A Pseudo code approach with C, Richard Gilberg ,Behrouz A. Forouzan, Cengage Learning.
4. Introduction to Data Structures in C- Ashok Kamthane, Pearson Education
5. Algorithms and Data Structures, Niklaus Wirth, Pearson Education

@The CO-PO mapping table												
PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	2	1	1	-	-	-	-	1	-	-	-
CO2	3	2	2	3	-	-	-	-	1	-	-	-
CO3	3	2	3	2	-	-	-	-	1	-	-	-
CO4	3	2	2	2	-	-	-	-	1	-	-	-
CO5	2	1	1	1	-	-	-	-	1	-	-	-
CO6	3	2	3	2	-	-	-	-	1	-	-	-

BSC-CA-301 Data Structures Lab		
Teaching Scheme **Practical: 04 Hours/Week**		**Examination Scheme and Marks** **Internal: 40 Marks** **External: 60 Marks**
Companion Course: Data Structures		
Course Objectives: ● To understand the standard and abstract data representation methods. ● To acquaint with the structural constraints and advantages in usage of the data. ● To understand the memory requirement for various data structures. ● To operate on the various structured data.		
Course Outcomes: On completion of the course, learner will be able to– CO1: To demonstrate the usage of various structures in approaching the problem solution. CO2: Apply the algorithms to solve the programming problems. CO3: Apply and analyze effective and efficient data structures in solving various Computer domain problems. CO4: Analyze the problems to apply suitable algorithm and data structure.		
Guidelines for Instructor's Manual The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..		

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended :- Windows / Linux

Programming tools recommended: - Turbo C++, Open Source C++ Programming tool like G++/GCC

Virtual Laboratory:

- https://cse01-iiith.vlabs.ac.in/
- https://ds1-iiith.vlabs.ac.in/Introduction.html

Data Structures Lab

Suggested List of Laboratory Experiments/Assignments (6 assignments are compulsory)	
Sr. No.	**Group A**
1.	Write C++ program for storing matrix and perform a) Matrix Addition b) Matrix Subtraction c) Matrix Multiplication
2.	Write C++ program to store student roll no of a class enrolled for training program in array in random order. Write function for- a) Searching whether particular student enrolled for training program or not using linear search. b) Searching whether particular student enrolled for training program or not using binary search.
3.	Write C++ program to store percentage of students in array. Write function for sorting array of floating point numbers in ascending order using a) Selection Sort b) Bubble sort and display top five scores.
4.	Write C++ program to implement Singly Linked List & perform the listed operations on it: a) Insertion b) Deletion c) Display d) Update e) Search
5.	Write C++ program for conversion of infix form of expression to postfix form.
6.	Write C++ program for implementation of Linear Queue using linked list.
7.	Write C++ program to perform the following: a) Create a binary search tree. b) Traverse the above binary search tree recursively in pre-order, post-order and in-order.
8.	Write C++ programs to implement the following graph traversal algorithms: a) Depth first search. b) Breadth first search
	Group B (Mini Project) **Select any one problem statement**
1	Implement tic-tac-toe.
2	Implement Snake game.
3	Build Cash Flow Analyzer
4	Implement Sudoku
5	Build Map Navigator

6	Implement File Zipper

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	2	-	-	-	-	-	-	-	-
CO2	3	3	2	2	-	-	-	-	-	-	-	-
CO3	3	3	2	2	-	-	-	-	-	-	-	-
CO4	2	3	2	2	-	-	-	-	-	-	-	

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year BCA (SEM 3) (2024-25 Course)

BCA-CA-302 Advanced Database Management Systems

Teaching Scheme:	Credit	Examination Scheme:
TH: 4 Hours/Week	**4**	**Internal (TH): 40 Marks** **External (TH): 60 Marks**

Prerequisite Courses, if any:

- Students must have fundamental knowledge of data structures & SQL queries.

Companion Course, if any: Database Management System

Course Objectives:

- To understand the fundamental concepts of Relational and Object-oriented databases.
- To learn and understand various Parallel and Distributed Database Architectures and Applications.
- To understand and apply the basic concepts, categories and tools of NoSQL Database
- To learn and understand Data warehouse and OLAP Architectures and Applications.
- To learn data mining architecture, algorithms, software tools and applications.
- To learn enhanced data models for advanced database applications.

Course Outcomes:

On completion of the course, learner will be able to–

CO1 Differentiate relational and object-oriented databases
CO2 Illustrate parallel & distributed database architectures.
CO3 Apply concepts of NoSQL Databases.
CO4 Explain concepts of data warehouse and OLAP technologies.
CO5 Apply data mining algorithms and various software tools
CO6 Comprehend emerging and enhanced data model for advanced applications.

Course Contents

Unit I	**Review Of Relational Data Model and Relational Database Constraints**	**(06 Hours)**

Relational model concepts, Relational model constraints and relational database schemas, Update operations, anomalies, dealing with constraint violations, Types and violations. **Overview of Object Oriented Concepts**–Objects, Basic properties. Advantages, examples, Abstract data types, Encapsulation, Class hierarchies, polymorphism examples

#Exemplar/Case Studies	Study and implement Polymorphism, Encapsulation
Mapping of Course Outcomes for Unit I	CO1

Unit II	**Concepts for Object Databases**	**(06 Hours)**

Object Identity– Object structure Type Constructors– Encapsulation of Operations – Methods – Persistence – Type and Class Hierarchies – Inheritance. XML Databases: XML - Related Technologies - XML Schema - XML Query Languages - Storing XML in Databases- XML and SQL.

#Exemplar/Case Studies	To implement XML in database
Mapping of Course Outcomes for Unit II	CO2

Unit III	**NOSQLDATABASES**	**(06 Hours)**

Introduction, Overview, and History of NoSQL Databases- The definition of Four Types of No SQL Databases. NoSQL Key/Value Database: MongoDB, Column-Oriented Database: Apache Cassandra, Comparison of Relational and NoSQL databases, NoSQL database Development Tools(Map Reduce/Hive) and Programming Languages(XML/JSON)

#Exemplar/Case Studies	To study and maintain the No SQL Database	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**DATA WAREHOUSING**	**(06 Hours)**
Architectures and components of data warehouse, Characteristics and limitations of data warehouse, Data ware house schema(Star, Snow flake), OLAP Architecture (ROLAP/MOLAP/HOLAP), Introduction to decision support system, Views and Decision support		
#Exemplar/Case Studies	Prepare and study Data Warehousing scheme	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**DATA MINING**	**(06 Hours)**
Transaction scheduling, serializability, Coping with System Failure, Concurrency Control techniques with locking, timestamp ordering and multiversion, Redo and Undo log based recovery, recovery in multi database systems		
#Exemplar/Case Studies	Prepare and Study Data Warehousing scheme	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**ENHANCED DATA MODELS FOR ADVANCED APPLICATIONS**	**(06 Hours)**
Active database concepts and triggers; Temporal, Spatial, and Deductive Databases–Basic concepts. More Recent Applications: Mobile databases; Multimedia databases; Geographical Information Systems; Genome data management.		
#Exemplar/Case Studies	Advanced models for data management	
Mapping of Course Outcomes for Unit VI	CO6	
Learning Resources		
Text Books: **3.** Silberschatz A.,Korth H., Sudarshan S, Database System Concepts,McGraw Hill Publication,ISBN-0- 07-120413-X,SixthEdition **4.** S. K. Singh, Database Systems: Concepts, Design and Application, Pearson Publication,ISBN-978- 81-317-6092-5		

Reference Books:

6. Kristina Chodorow, Michael Dirolf,—MongoDB: The Definitive Guide‖,O'Reilly Publications
7. Jiawei Han,MichelineKamber,JianPei,—Data Mining: Concepts and Techniques‖ , Elsevier
8. Mario Piattini ,Oscar Diaz—Advanced Database Technology and Design‖-online book.
9. M. Tamer Özsu, Patrick Valduriez, —Principles of Distributed Database Systems‖ PrenticeHall,1999.
10. Ramez Elmasri and Shamk ant B. Navathe —Fundamentals of Database System‖7 th Edition

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	2	1	-	-	-	-	-	-	-	-
CO2	1	2	-	2	-	-	-	-	-	-	-	-
CO3	2	1	2	1	-	-	-	-	-	-	-	-
CO4	1	2	-	-	-	-	-	-	-	-	-	-
CO5	-	-	2	-	-	-	-	-	-	-	-	-
CO6	-	2	1	2	-	-	-	-	-	-	-	-

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

BCA (2024-25 Course)

BCA-CA-303: Operating System Principles

Teaching Scheme:	Credit	Examination Scheme:
TH: 3 Hours/Week	**5**	**Internal (TH): 40 Marks** **External (TH): 60 Marks**

Prerequisite Courses, if any:

- Programming Languages.
- Data Structures and Algorithms.

Companion Course, if any: System Software

Course Objectives:

- To explain main components of OS and their working.
- To familiarize the operations performed by OS as a resource Manager.
- To impart various scheduling policies of OS.
- To teach the different memory management techniques.

Course Outcomes:

On completion of the course, learner will be able to–

CO7: Outline the basic concept of operating systems.
CO8: Analyze the working of operating system.
CO9: Examine the working of various scheduling/allocation approaches.
CO10: Measure the performance of various scheduling/allocation approaches.
CO5: Implement algorithm of CPU Scheduling, Memory Scheduling and disk scheduling.
CO6: Compare various operating systems with respect to characteristics and features.

Course Contents

Unit I	**OPERATING SYSTEMS**	**(07 Hours)**
OPERATING SYSTEMS OVERVIEW: Introduction, operating system operations, process management, memory management, storage management, protection and security, distributed systems. OPERATING SYSTEMS STRUCTURES: Operating system services and systems calls, system programs, operating system structure, operating systems generations.		
#Exemplar/Case Studies	Case Study: Introduction to OS concepts using Windows 10. Example: Exploring file management, user interfaces, and basic system utilities.	
Mapping of Course Outcomes for Unit I	Outline the basic concept of operating systems	
Unit II	**PROCESS MANAGEMENT**	**(08 Hours)**
PROCESS MANAGEMENT: Process concepts, process state, process control block, scheduling queues, process scheduling, multithreaded programming, threads in UNIX, comparison of UNIX and windows.		
#Exemplar/Case Studies	Case Study: Managing concurrent processes in Linux. Example: Implementing semaphores and mutexes in a multi-threaded application to prevent race conditions.	
Mapping of Course Outcomes for Unit II	Analyze the working of operating system	
Unit III	**CONCURRENCY AND SYNCHRONIZATION**	**(07 Hours)**

CONCURRENCY AND SYNCHRONIZATION: Process synchronization, critical section problem, Peterson's solution, synchronization hardware, semaphores, classic problems of synchronization, readers and writers problem, dining philosophers problem, monitors, synchronization examples(Solaris), atomic transactions. Comparison of UNIX and windows.		
#Exemplar/Case Studies	Case Study: Avoiding deadlocks in database systems. Example: Implementing deadlock detection algorithms and resource allocation graphs in a transaction management system.	
Mapping of Course Outcomes for Unit III	Measure the performance of various scheduling/allocation approaches	
Unit IV	**DEADLOCKS**	**(08 Hours)**
DEADLOCKS: System model, deadlock characterization, deadlock prevention, detection and avoidance, recovery from deadlock banker's algorithm. MEMORY MANAGEMENT: Swapping, contiguous memory allocation, paging, structure of the page table, segmentation, virtual memory, demand paging, page-replacement algorithms, allocation of frames, thrashing, case study - UNIX.		
#Exemplar/Case Studies	Case Study: Ensuring data consistency in multi-user environments. Example: Using locking mechanisms and isolation levels in SQL Server to manage concurrent access to data.	
Mapping of Course Outcomes for Unit IV	Implement algorithm of CPU Scheduling, Memory Scheduling and disk scheduling.	
Unit V	**FILE SYSTEM**	**(06 Hours)**
FILE SYSTEM: Concept of a file, access methods, directory structure, file system mounting, file sharing, protection. File system implementation: file system structure, file system implementation, directory implementation, allocation methods, free-space management, efficiency and performance, comparison of UNIX and windows.		
#Exemplar/Case Studies	Case Study: Managing file storage and access in Unix-based systems.	
Mapping of Course Outcomes for Unit V	Example: Implementing file permissions, directory structures, and inodes in an Ext4 file system.	
Unit VI	**I/O SYSTEM**	**(06 Hours)**
I/O SYSTEM: Mass storage structure - overview of mass storage structure, disk structure, disk attachment, disk scheduling algorithms, swap space management, stable storage implementation, tertiary storage structure. I/O: Hardware, application I/O interface, kernel I/O subsystem, transforming I/O requests to hardware operations, streams, performance.		

#Exemplar/Case Studies	Case Study: Enhancing I/O operations in embedded systems. Example: Using interrupt handling and buffering techniques to improve data transfer efficiency in real-time applications.
Mapping of Course Outcomes for Unit VI	Compare various operating systems with respect to characteristics and features

Learning Resources

Text Books:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne (2006), Operating System Principles, 7th edition, Wiley India Private Limited, New Delhi.

Reference Books:

1. Stallings (2006), Operating Systems, Internals and Design Principles, 5th edition, Pearson Education, India.

2. Andrew S. Tanenbaum (2007), Modern Operating Systems, 2nd edition, Prentice Hall of India, India.

3. Deitel & Deitel (2008), Operating systems, 3rd edition, Pearson Education, India.

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	3	1	-	-	-	-	1	-	2	2
CO2	3	3	3	1	-	-	-	-	1	-	1	1
CO3	3	2	3	1	-	-	-	-	1	-	-	-
CO4	3	2	3	1	1	-	-	-	1	-	-	-
CO5	3	2	3	1	1	-	-	-	1	-	-	1
CO6	3	2	3	1	1	-	-	-	1	-	-	1

Operating System Principles Lab		
Teaching Scheme **Practical: 02 Hours/Week**		**Examination Scheme and Marks** **Internal: 40 Marks** **External: 60 Marks**

Companion Course:

Course Objectives:

- To explain main components of OS and their working.
- To familiarize the operations performed by OS as a resource Manager.
- To impart various scheduling policies of OS.
- To teach the different memory management techniques.

Course Outcomes:

CO1Implementation of various scheduling/allocation approaches
CO2 Measure the performance of various scheduling/allocation approaches through program.
CO3: Implement algorithm of CPU Scheduling, Memory Scheduling and disk scheduling.
CO4: Compare various operating systems Algorithm for multitasking

Virtual Laboratory:

-

Operating System & Principles Lab

Suggested List of Laboratory Experiments/Assignments

Sr. No.	Group A
9.	Develop a basic command-line shell that can execute simple commands.
10.	Create a basic file system to support file creation, deletion, and reading.
11.	Implement a simple process scheduler using Round Robin scheduling.
12.	Simulate basic memory allocation techniques like first-fit and best-fit.
13.	Develop a basic semaphore to manage access to a shared resource.
14.	Implement a basic page replacement algorithm (e.g., FIFO).
15.	Create a simple tool to detect deadlocks using a resource allocation graph.
16.	Develop a simple tool to monitor and display CPU usage.
	Group B (Mini Project) **Select any one problem statement**
7	Create a basic system for user authentication and login.
8	Implement a basic network packet sniffer to capture network packets.
9	Develop a basic RAM disk to create a temporary file system in RAM.
1	Implement basic threading operations like thread creation and termination.
1	Simulate simple disk scheduling algorithms like FCFS.

	1	Create a tool to visualize resource allocation and detect potential deadlocks.
	1	Implement a basic power management system to simulate different power modes.
	1	Develop a basic loadable kernel module that adds simple functionality.

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	2	2	3	-	-	-	-	-	-	1
CO2	1	2	2	2	2	-	-	-	-	-	-	1
CO3	1	2	2	2	2	-	-	-	-	-	-	1
CO4	-	2	-	2	1	-	-	-	-	-	-	-

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year of Engineering (2024-25 Course)

PCC-CA-301: Project Management

Teaching Scheme:	Credit	Examination Scheme:
TH: 01 Hours/Week	02	**Internal (TH): 20 Marks** **External (TH): 30 Marks**

Prerequisite Courses, if any:

- Students must have a knowledge of fundamentals of software Engineering

Companion Course, if any:

Course Objectives:

Course Objective:

- To learn and understand the principles of Project Management.
- To be acquainted with methods of Project Life cycle
- To apply Design and Testing principles to project development.
- To understand project management through life cycle of the project.

Course Outcomes:

CO1: Understand the concepts of project management.
CO2: Understand the Project life cycle.
CO3: Create a project schedule using various tools.
CO4: Estimate the project cost.
CO5: Explain the Project Communication Management.
CO6: Explain various human resource planning.

Course Contents

Unit I	**Introduction to Project Management**	**(02Hours)**
Knowledge areas as per PMBOK, Project Scope Management, Project Charter and Stakeholder Management		
#Exemplar/Case Studies	**The Sydney Opera House Project**	
Mapping of Course Outcomes for Unit I	CO1	
Unit II	**Project Life Cycle & Initiation**	**(02Hours)**
Project Life Cycle & Initiation, Portfolio Approach to Project Management, Project/Portfolio Selection & Organizational Strategy, Project Planning		
#Exemplar/Case Studies	**The Airbus A380 Project**	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**Project Scheduling & Risk Analysis**	**(02Hours)**
Project Scheduling, Project Cost Management, Risk Analysis in Project Management, Exposure to Software applicable in Project Management		
#Exemplar/Case Studies	**The Apple iPhone Development Project**	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**Project Procurement**	**(02Hours)**

Project Procurement and Supply Chain Management, Project Quality Management, Six Sigma & Project Management, Critical Chain Project Management		
#Exemplar/Case Studies	**The Apple iPhone Development Project**	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Project Communication Management**	**(02Hours)**
Project Communication Management, Software Project Management and Adaptive & Agile Project Management, PM Process Framework and Value Delivery Systems in Project Management, Behavioral & Leadership aspects of Project Management		
#Exemplar/Case Studies	**The Tesla Electric Car Project**	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**Human Resource Planning**	**(02Hours)**
Human Resource Planning in Project Management, Business Analytics, AI and Automation in Project Management, Project Commissioning, Closure & Handover		
#Exemplar/Case Studies	**Online Marketplace Platform Project**	
Mapping of Course Outcomes for Unit VI	CO6	
Learning Resources		
Text Books: 1. Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 10th ed.		

Reference Books:

1. Project Management Absolute Beginner's Guide Series, Greg Horine, illustrated, reprint, Que, 2013, 0789750104, 9780789750105
2. Making Things Happen: Mastering Project Management By Scott Berkun
3. Strategic Project Management Made Simple: Practical Tools for Leaders and Teams
   Strategic Project Management Made Simple: Practical Tools for Leaders and Teams, Terry Schmidt, John Wiley & Sons, 2009, ISBN : 047044293X, 9780470442937

@The CO-PO mapping table

CO\ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	2	-	-	-	-	-	-	-	-	-	-	-
CO2	2	1	-	-	2	-	-	-	-	-	-	1
CO3	2	-	-	-	1	-	-	-	1	-	1	-
CO4	2	-	-	-	-	-	-	-	-	-	-	-
CO5	2	-	-	-	1	-	-	-	-	-	-	1
CO6	2	-	-	-	-	-	-	-	-	-	-	-

PCC-CA-301: Project Management Lab

Teaching Scheme **Practical: 02 Hours/Week**		**Examination Scheme and Marks** **Internal: 20 Marks** **External: 30 Marks**

Companion Course:

Course Objectives:

- Apply various software engineering concepts for real world applications.
- Apply various project management concepts for real world applications.

Course Outcomes:

On completion of the course, learner will be able to–

- **CO1:** Understand real world problem statements.
- **CO2**: Create project schedule.
- **CO3**: Understand and apply the Project testing concepts.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended :- Windows

Programming tools recommended: - Jira,Height

Virtual Laboratory:

PCC-CA-301: Project Management Lab	
Suggested List of Laboratory Experiments/Assignments **(6 assignments are compulsory)**	
Sr. No.	**Group A**
17.	Problem Identification and justification
18.	Feasibility study of the project to the organization
19.	Preparation of Statement of Work
20.	Create Work Breakdown structure using Gantt chart
21.	Project budget and cost distribution plan
22.	Communications Management Plan
23.	Quality control plan for the project.
	Group B (Mini Project) **Select any one problem statement**
1	Online hotel booking systems
1	Stock Market Risk Analysis
1	Hospital Management System
1	Shopping Mall Inventory Management
1	Student Attendance Management System
2	Restaurant Management system
2	Railway reservation system

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	2	2	3	-	-	-	-	-	-	1

CO2	1	2	2	2	2	-	-	-	-	-	-	1
CO3	1	2	2	2	2	-	-	-	-	-	-	1

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year of BCA (2024-25 Bachelor of Computer Application)

PEC- CA-301A: Discipline Specific Elective -3

Teaching Scheme:	**Credit**	**Examination Scheme:**
TH: 2 Hours/Week	**4**	**Internal (TH): 40 Marks** **External (TH): 60 Marks**

Prerequisite Courses, if any:

- Basic of computer literacy
- Fundamentals of HTML and CSS

Companion Course, if any: **Programming in Python**

Course Objectives:

- Learn the syntax and semantics of the Python programming language.
- Illustrate the process of structuring the data using lists, tuples
- Appraise the need for working with various documents like Excel, PDF, Word and Others.
- Demonstrate the use of built-in functions to navigate the file system.
- Implement the Object Oriented Programming concepts in Python.

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Demonstrate proficiency in handling loops and creation of functions.
CO2: Identify the methods to create and manipulate lists, tuples and dictionaries.
CO3: Develop programs for string processing and file organization
CO4: Interpret the concepts of Object-Oriented Programming as used in Python.
CO5: Manipulate file system in Python.
CO6: Handle errors and exception in Python applications.

Course Contents

Unit I	Python Basics and Flow control	(07 Hours)

Python Basics: Entering Expressions into the Interactive Shell, The Integer, Floating-Point, and String Data Types, String Concatenation and Replication, Storing Values in Variables, Your First Program, Dissecting Your Program.

Flow control: Boolean Values, Comparison Operators, Boolean Operators, Mixing Boolean and Comparison Operators, Elements of Flow Control, Program Execution, Flow Control Statements, Importing Modules, Ending a Program Early with sys. exit().

#Exemplar/Case Studies	Use of flow control for Arithmetic and Boolean operation
Mapping of Course Outcomes for Unit I	CO1

Unit II	Functions	(08 Hours)

def Statements with Parameters, Return Values and return Statements, The None Value, Keyword Arguments and print(), Local and Global Scope, The global Statement, Exception Handling, A Short Program: Guess the Number.

#Exemplar/Case Studies	Use of different statement
Mapping of Course Outcomes for Unit II	CO2

Unit III	Lists, Dictionaries and Structuring Data	(07 Hours)

Lists: The List Data Type, Working with Lists, Augmented Assignment Operators, Methods, Example Program: Magic 8 Ball with a List, List-like Types: Strings and Tuples, References.

Dictionaries and Structuring Data: The Dictionary Data Type, Pretty Printing, Using Data Structures to Model Real-World Things.

#Exemplar/Case Studies	Use of different Data and Operator

Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**Manipulating Strings, function and Project**	**(08 Hours)**
Manipulating Strings: Working with Strings, Useful String Methods, Project: Password Locker, Project: Adding Bullets to Wiki Markup Reading and Writing Files: Files and File Paths, The os.path Module, The File Reading/Writing Process, Saving Variables with the shelve Module, Saving Variables with the print.format(). **Function, Project:** Generating Random Quiz Files, Project: Multiclipboard.		
#Exemplar/Case Studies	Use of different string and variables	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Organizing Files and Debugging**	**(06 Hours)**
Organizing Files: The shutil Module, Walking a Directory Tree, Compressing Files with the zipfile Module, Project: Renaming Files with American-Style Dates to European-Style Dates,Project: Backing Up a Folder into a ZIP File. **Debugging:** Raising Exceptions, Getting the Traceback as a String, Assertions, Logging, IDLE‟s Debugger.		
#Exemplar/Case Studies	Operating files and debugging	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**Functions and Methods of Classes and Objects**	**(06 Hours)**
Classes and objects: Programmer-defined types, Attributes, Rectangles, Instances as return values, Objects are mutable, Copying. **Classes and functions:** Time, Pure functions, Modifiers, Prototyping versus planning. **Classes and methods:** Object-oriented features, Printing objects, Another example, A more complicated example, Theinit method, The __str__ method, Operator overloading, Type-based dispatch, Polymorphism, Interface and implementation.		
#Exemplar/Case Studies	Use of classes and objects	
Mapping of Course Outcomes for Unit VI	CO6	
Learning Resources		

Text Books:

1. Al Sweigart,"Automate the Boring Stuff with Python",1 stEdition, No Starch Press, 2015. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/) (Chapters 1 to 18, except 12) for lambda functions use this link: https://www.learnbyexample.org/python-lambda-function/
2. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist", 2 nd Edition, Green Tea Press, 2015. (Available under CC-BY-NC license at http://greenteapress.com/thinkpython2/thinkpython2.pdf (Chapters 13, 15, 16, 17, 18) (Download pdf/html files from the above link)

Reference Books:

1. Downey, A. et al., "How to think like a Computer Scientist: Learning with Python", John Wiley, 2015
2. Mark Lutz, "Learning Python", 5th edition, Orelly Publication, 2013, ISBN 978- 1449355739
3. John Zelle, "Python Programming: An Introduction to Computer Science", Second edition, Course Technology Cengage Learning Publications, 2013, ISBN 978- 1590282410
4. Michel Dawson, "Python Programming for Absolute Beginers" , Third Edition, Course Technology Cengage Learning Publications, 2013, ISBN 978-1435455009
5. David Beazley, Brian Jones., "Python Cookbook", Third Edition, Orelly Publication, 2013, ISBN 978-1449340377
6. Martin C. Brown, Python: The Complete Reference, Osborne/McHraw Hill, 2001

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	-	-	-	-	-	-	-	-	-	-	-
CO2	3	-	-	-	-	-	-	-	-	-	-	-
CO3	3	3	-	-	-	-	-	-	-	-	-	-
CO4	-	-	-	-	-	-	-	-	-	-	-	-
CO5	3	-	-	-	-	-	-	-	-	-	-	-
CO6	3	3	-	-	-	-	-	-	-	-	-	-

PEC- CA-301A: Discipline Specific Elective -3

Teaching Scheme Practical: 04 Hours/Week		Examination Scheme and Marks Internal: 40 Marks External: 60 Marks

Companion Course: Programming in Python

Course Objectives:

- To implement the python programming features in practical applications.
- To write, test, and debug simple Python programs.
- To implement Python programs with conditionals and loops.
- Use functions for structuring Python programs.
- Represent compound data using Python lists, tuples, dictionaries and modules

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Develop algorithmic solutions to simple computational problems
CO2: Develop and execute simple Python programs.
CO3: Implement programs in Python using conditionals and loops for solving problems.
CO4: Deploy functions to decompose a Python program
CO5: Process compound data using Python data structures.
CO6: Utilize Python packages in developing software applications.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended :- Unix, Linux, Microsoft Windows, macOS, Darwin

Programming tools recommended: - Pip, Jupyter Notebook, Anaconda, Visual Studio Code

Virtual Laboratory:

- NPTEL & MOOC courses titled Python programming
- https://spoken-tutorial.org/tutorial-search/?search_foss=Python&search_language=English
- https://docs.python.org/3/tutorial/index.html
- https://runestone.academy/runestone/static/pythonds

Part I : Programming in Python

Suggested List of Laboratory Experiments/Assignments

(6 assignments are compulsory)

Sr. No.	Group A
24.	Program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon user's choice.
25.	Program to calculate total marks, percentage and grade of a student. Marks obtained in each of the five subjects are to be input by user. Assign grades according to the following criteria: Grade A: Percentage >=80 Grade B: Percentage >=70 and <80 Grade C: Percentage >=60 and <70 Grade D: Percentage >=40 and <60 Grade E: Percentage <40
26.	Program, using user-defined function to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user
27.	Program to display the first n terms of Fibonacci series.

28.	Program to find factorial of the given number.
29.	Write a Python program to count the number of even and odd numbers from N numbers
30.	Python function that accepts a string and calculate the number of upper case letters and lower case letters.
31.	Python program to reverse a given string and check whether the give string palindrome or not.
32.	Write a program to find sum of all items in a dictionary.
33.	Write a Python program to construct the following pattern, using a nested loop 1 22 333 4444 55555 666666 7777777 88888888 999999999
	Group B (Mini Project) Select any one problem statement
2	Build a Simple Auto-Login Bot with Python
2	Build an Automated Employee Management System
2	Predict Air Quality Index using Python
2	Build an Automated software testing with Python
2	Create program to send automated email messages which involve delivering text messages, essential photos, and important files, among other things.
2	Create program to Convert PDF File Text to Audio Speech using Python
2	Build a Text detection using Python
2	Build a Simple Attendance Tracker using Python

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	3	2	-	-	-	-	-	2	2

CO2	3	3	3	3	2	-	-	-	-	-	2	2
CO3	3	3	3	3	2	-	-	-	-	-	2	-
CO4	2	2	-	2	2	-	-	-	-	-	1	-
CO5	1	2	-	-	1	-	-	-	-	-	1	-
CO6	2	2	-	-	2	-	-	-	-	-	1	_

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year BCA (2024-25 Bachelor of Computer Applications)

PEC- CA-301B: Discipline Specific Elective -3

Teaching Scheme:	Credit	Examination Scheme:
TH: 2 Hours/Week	4	Internal (TH): 40 Marks External (TH): 60 Marks

Prerequisite Courses, if any:

- Basic of Programming.
- Fundamentals of C++ and Object Oriented Programming

Companion Course, if any: Java

Course Objectives:

- To learn the various features of Java and comparing with C++.
- To learn the Java environment for writing programs and Java program structure
- To learn the various Objects oriented features with Java.
- To learn the Array and String concepts in Java.
- To learn the method of Exception Handling in Java.
- To learn the concepts of Thread and Package.
- To learn the Applet concepts and implementing them in creating a web page.

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Able to understand the features of Java Programming Language with Syntax and structure of Java Programs and how to use various operators in Java.
CO2: Able to understand that how to implement the Object oriented features by writing Java programs
CO3: Ability to define Arrays, Strings, Vectors, Packages etc. in Java and implementing the Exception handling Mechanism in Java.
CO4: Ability to understand the different concepts to create and use Threads and Packages in Java.
CO5: Ability to understand the different concepts of applets and adding them to a HTML File.

Course Contents

Unit I	An Overview of Java	(08 Hours)

Object-Oriented Programming, A First Simple Program, A Second Short Program, Two Control Statements, Using Blocks of Code, Lexical Issues, The Java Class Libraries, Data Types, Variables, and Arrays: Java Is a Strongly Typed Language, The Primitive Types, Integers, Floating-Point Types, Characters, Booleans, A Closer Look at Literals, Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays, A Few Words About Strings		
#Exemplar/Case Studies	Use of Object Oriented Programming	
Mapping of Course Outcomes for Unit I	CO1	
Unit II	**Operators**	**(08 Hours)**
Arithmetic Operators, The Bitwise Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses, Control Statements: Java‟s Selection Statements, Iteration Statements, Jump Statements.		
#Exemplar/Case Studies	Use of different operator	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**Introducing Classes**	**(06 Hours)**
Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection, The finalize( ) Method, A Stack Class, A Closer Look at Methods and Classes: Overloading Methods, Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects, Recursion, Introducing Access Control, Understanding static, Introducing final, Arrays Revisited		
#Exemplar/Case Studies	Use of different classes	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**Inheritance**	**(08 Hours)**
Inheritance, Using super, Creating a Multilevel Hierarchy, When Constructors Are Called, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, The Object Class.		
#Exemplar/Case Studies	Text book 1: Ch 8	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Packages and Interfaces**	**(06 Hours)**

Packages, Access Protection, Importing Packages, Interfaces, Exception Handling: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java‟s Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions, Using Exceptions.

#Exemplar/Case Studies	Use of different packages and interface
Mapping of Course Outcomes for Unit V	CO5

Learning Resources

Text Books:

1. Herbert Schildt, Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2007

Reference Books:

1. Programming with JAVA - E Balgurusamy
2. The Complete Reference – JAVA Herbert Schildt

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1		1	1	-	-	-	-	-	-	-
CO2	2	-	-	-	1	2	-	2	-	-	-	-
CO3	2	1	3	1	-	1	2	1	-	-	-	-
CO4	1	1	2	1	-	3	1	2	-	-	-	-
CO5	1	1	2	1	-	2	1	-	-	-	-	-

PEC- CA-301B: Discipline Specific Elective -3 Lab

Teaching Scheme		**Examination Scheme and Marks**
Practical: 04 Hours/Week		**Internal: 40 Marks** **External: 50 Marks**

Companion Course: JAVA

Course Objectives:

- Demonstrate Object oriented constructs such as various class hierarchies, interfaces and Packages.

- Develop and understand Exception handling.
- To understand the concepts of threads and I/O in Java.
- Able to build dynamic user interfaces using applets and Event handling in java.

Course Outcomes:

On completion of the course, learner will be able to–

- **CO1:** Make use of Class, inheritance and interface
- **CO2:** Create package, user defined excepts and treads
- **CO3:** Construct a frame for various controls
- **CO4:** Construct applet for multiple shapes
- **CO5:** Test various files in java

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended :- Windows

Programming tools recommended: - VisualVM, Git

Virtual Laboratory:

- https://onlinecourses.nptel.ac.in/noc22_cs47/preview

Part I : JAVA Lab

Suggested List of Laboratory Experiments/Assignments

(6 assignments are compulsory)

Sr. No.	Group A(Two Assignments are compulsory)
1.	Write a JAVA program that prints all real solutions to the quadratic equation ax2+bx+c=0. Read in a, b, c and use the quadratic formula.
2.	Write a JAVA program for multiplication of two arrays.
3.	Demonstrate the following operations and sign extension with Java programs (i) << (ii) >> (iii) >>>
4.	Write aJAVA program to sort list of elements in ascending and descending order
5.	Create a JAVA class called Student with the following details as variables within it. PRN NAME BRANCH PHONE PERCENTAGE Write a JAVA program to create n Student objects and print the USN, Name, Branch, Phone, and percentage of these objects with suitable headings.
6.	Write a JAVA program demonstrating Method overloading and Constructor overloading.

7.	Design a super class called Staff with details as StaffId, Name, Phone, and Salary. Extend this class by writing three subclasses namely Teaching (domain, publications), Technical (skills), and Contract (period). Write a JAVA program to read and display at least 3 staff objects of all three categories.
8.	Demonstrate dynamic dispatch using abstract class in JAVA.
9.	Create two packages P1 and P2. In package P1, create class A, class B inherited from A, class C. In package P2, create class D inherited from class A in package P1 and class E. Demonstrate working of access modifiers (private, public, protected, default) in all these classes using JAVA.
10.	Write a JAVA program to read two integers a and b. Compute a/b and print, when b is not zero. Raise an exception when b is equal to zero. Also demonstrate working of Array Index Out Of Bound Exception.
	Group B (Mini Project) Select any one problem statement
1.	Electricity Billing System
2.	Web Medical Management System
3.	Supply Chain Management System
4.	Exam Seating Arrangement System in Java
5.	Create a Criminal Face Detection System
6.	Teachers Feedback Form Java Project
7.	Online Library Management System
8.	Vehicle Identification System

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1		1	1	-	-	-	-	-	-	-
CO2	2	-	-	-	1	2	-	2	-	-	-	-
CO3	2	1	3	1	-	1	2	1	-	-	-	-
CO4	1	1	2	1	-	3	1	2	-	-	-	-
CO5	1	1	2	1	-	2	1	-	-	-	-	-

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year of BCA (SEM 3)(2024-25)

PEC- CA-301C: C#

Teaching Scheme:	**Credit**	**Examination Scheme:**
TH: 2 Hours/Week	**04**	**Internal (TH): 40 Marks** **External (TH): 60 Marks**

Prerequisite Courses, if any:

- To understand and learn the concepts of C# and .Net Programming
- To implement various live project

Companion Course, if any: Web Technology

Course Objectives:

- To understand and learn the concepts of C# and .Net Programming
- To implement various live project

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Able to explain how C# fits into the .NET platform.

CO2: Describe the utilization of variables and constants of C#.

CO3: Use the implementation of object-oriented aspects in applications.

CO4: Analyze and Set up Environment of .NET Core.

CO5: Evaluate and create a simple project application.

Course Contents

Unit I	**.Net Framework Overview**	**(07 Hours)**
Architecture-.Net Framework class Libraries-CLR-Metadata-Interoperability-Assemblies-the .net Packaging system-CLR-MSIL , Introduction to Visual Studio.Net-C# Programming Concepts-Predefined Types- Value types and reference type, Classes and Objects, Constructors and methods , Conditional statements, loops, arrays , Collection classes: ArrayList , HashTable, Stack ,Queue, indexers and properties.		
#Exemplar/Case Studies	Book Call Number, Dictionary,	
Mapping of Course Outcomes for Unit I	CO1	
Unit II	**String class**	**(08 Hours)**
Methods and properties of string class, enumerations, boxing and unboxing, OOPS concepts: Encapsulation, data hiding, inheritance, interfaces, polymorphism, operator overloading, overriding Methods, Static Class members, Delegates and events. Exception Handling, garbage collector, generics and collection		
#Exemplar/Case Studies	Basics of OOPs concepts	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**Basics of Windows Programming**	**(07 Hours)**
Event Driven Programming, Windows Forms, Using common controls-Labels, textboxes, buttons, check boxes, radio button, progress bar, combo box, list box. Components-timer, imagelist, Menus, Modal and Modeless Dialog Boxes, MDI, Mouse and keyboard event handling.		
#Exemplar/Case Studies	Use of Functions	

Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**Introduction to ADO.Net-Object Model- System**	**(08 Hours)**
Data Namespace- Data Bound controls- Connected Mechanism-Disconnected mechanism-.Net Data Providers.		
#Exemplar/Case Studies	Use of mechanism	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Files**	**(06 Hours)**
System.IO, directory and file types, Stream readers and stream writers, working with binary data.		
#Exemplar/Case Studies	Use of different Windows and Tools	
Mapping of Course Outcomes for Unit V	CO5	

Learning Resources

Text Books:

1. C# 4.0 the Complete Reference by Herbert Schildt

Reference Books:

1. Latest version of Andrew Trolsens C# text from Apress(Pro C# 5.0 and the .NET Framework 4.5)
2. Robert Powel, Richard Weeks, C# and the .NET Framework, Techmedia

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	-	-	2	-	-	-	-	-	-	2
CO2	3	3	3	-	2	-	1	1	2	-	1	-
CO3	3	3	3	-	2	1	1	1	2	-	1	-
CO4	1	1	3	-	2	1	1	-	-	-	-	-
CO5	3	3	3	-	2	1	1	-	-	-	-	1

PEC- CA-301C: C# Lab		
Teaching Scheme **Practical: 04 Hours/Week**		**Examination Scheme and Marks** **Internal: 40 Marks** **External: 60 Marks**

Companion Course: C#

Course Objectives:

- To learn basic features of C# programming
- To understand C# support for OOP with programming examples
- To gain experience of modern tool usage (VS Code, Visual Studio or any other] in developing C# programs

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Develop programs involving basic features of C# programming language

CO2: Make use of exception handling features to safeguard program against runtime anomalies

CO3: Apply concepts of OOP in developing solutions to problems

CO4: Develop programs to illustrate handling of text files

CO5: Make use of modern tools to develop C# programs and applications

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended :- Windows

Programming tools recommended: - Visual Studio , VS Code , or Command Line

Virtual Laboratory:

- https://www.tutorialsteacher.com/csharp
- https://www.w3schools.com/cs/index.php
- https://www.javatpoint.com/net-framework

Part I : C# Lab

Suggested List of Laboratory Experiments/Assignments

(6 assignments are compulsory)

Sr. No.	Group A
1.	Develop a C# program to simulate simple arithmetic calculator for Addition, Subtraction, Multiplication, Division and Mod operations. Read the operator and operands through console.
2.	Develop a C# program to print Armstrong Number between 1 to 1000.
3.	Develop a C# program to list all substrings in a given string. [Hint: use of Substring() method]
4.	Develop a C# program to demonstrate Division by Zero and Index Out of Range exceptions.

5.	Develop a C# program to generate and printPascal Triangle using Two Dimensional arrays.
6.	Develop a C# program to generate and print Floyds Triangle using Jagged arrays.
7.	Develop a C# program to read a text file and copy the file contents to another text file.
8.	Develop a C# C# Program to Implement Stack with Push and Pop Operations [Hint: Use class, get/set properties, methods for push and pop and main method]
	Group B (Mini Project) Select any one problem statement
1.	Simple Billing System
2.	Simple ATM System
3.	Student Grading System
4.	Online Cinema Ticket Booking System
5.	Pharmacy Management System
6.	Employee Management System Project
7.	Exam Scheduler System
8.	SMS Based Remote Server Monitoring System

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	1	1	2	1	-	-	1	1	-	2
CO2	2	2	1	1	2	-	-	-	1	-	1	2
CO3	2	2	2	2	1	1	-	-	2	-	-	2
CO4	2	2	1	1	3	-	-	-	1	1	-	2
CO5	2	2	2	2	2	1	-	-	2	-	-	1

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year BCA (SEM 3) (2024-25)

PEC- CA-301D: Ruby

Teaching Scheme:	**Credit**	**Examination Scheme:**
TH: 2 Hours/Week	**4**	**Internal (TH): 40 Marks** **External (TH): 60 Marks**

Prerequisite Courses, if any:

- Understanding of its basics is essential.
- Object-Oriented Programming (OOP): Since Ruby is a purely Object-Oriented Language, a basic understanding of OOP concepts would be helpful.
- Basic knowledge of HTML, CSS, and JS.

Companion Course, if any: Web Technology

Course Objectives:

- understand platform requirements for a robust development environment
- Connect to a Rails application and utilize Rails tools for improved productivity.
- Grasp the principles of the MVC architecture and apply the "Convention over Configuration" philosophy for efficient coding.
- Develop proficiency in Ruby by learning about objects, variables, methods, classes, modules, and built-in classes

Course Outcomes:

CO1: On completion of the course, learner will be able to–
CO2: Develop and test programs using the Ruby programming language.
CO3: Develop, test, and deploy basic web applications with Ruby on Rails (RoR).
CO4: Develop, test, and deploy web layout and user models using RoR.
CO5: Create an advanced project using MySQL, Ruby and the Ruby on Rails framework.

Course Contents

Unit I	**Introduction**	**(07 Hours)**
What is Ruby, Why ruby, General purpose of ruby, Brief History of Ruby, Where does ruby get its ideas, Ruby Installation with RVM, Installations of Software (RVM, Rails , GIT, Mysql, Ruby, Sublime Text Editior), Rvm Commands, Rvm Usage, Creating a basic script in ruby, Sample demo of ruby program.		
#Exemplar/Case Studies	Book Call Number, Dictionary	
Mapping of Course Outcomes for Unit I	CO1	
Unit II	**Basics of Ruby**	**(08 Hours)**
Hello, Matz, Interactive Ruby. Ruby Is Object-Oriented, Ruby's Reserved Words, Variables, Strings, Numbers and Operators. Conditional Love, The if Statement, The case Statement, The while Loop, The loop Method, The for loop. Strings , Creating Strings, Concatenating Strings, Accessing Strings, Comparing Strings, Manipulating Strings, Case Conversion, Managing Whitespace, Incrementing Strings, Converting Strings, Regular Expressions. Math, Class Hierarchy and Included Modules, Converting Numbers, Basic Math Operations, Math Methods, Math Functions		
#Exemplar/Case Studies	Use of different functions	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**Rails Installation and Ruby Gems**	**(07 Hours)**
What is Rails, Full tack Framework, Rails Strength, COC(convention over configuration),Rails Installation, Ruby on Rails installation on linux, Ruby Gems, Working with Ruby Gems, Gem commands Framework Technology MVC Rails Components.		
#Exemplar/Case Studies	Use of different operating system and tools	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**Arrays**	**(08 Hours)**

Creating Arrays, Accessing Elements, Concatenation, Set Operations, Comparing Arrays, Changing Elements, Deleting Elements, Multidimensional Arrays. Hashes, Creating Hashes, Accessing Hashes, Iterating over Hashes, Changing Hashes. Classes ,Defining the Class, Instance Variables, Accessors, Class Variables, Class Methods, Inheritance, Modules, public, private, or protected.		
#Exemplar/Case Studies	Use of arrays	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Programming Ruby and working Sample**	**(06 Hours)**
Programming Ruby: Defining methods, conditionals ,if/elsif/else/unless, ternary operator, case statement, looping – for/in loop, while and until loops, blocks and iterators, exception handling, raising errors, objects and classes, defining and instantiating classes, attributes and accessor methods, methods visibility, single inheritance, monkey patching, singleton methods and eigenclasses. A Working Sample: Creating a New Rails Application, Creating Databases, Scaffolding and Migrations, Putting It All Together: Creating a Rails Application.		
#Exemplar/Case Studies	Different algorithm	
Mapping of Course Outcomes for Unit V	CO5	
Learning Resources		
Text Books: 1. Learn Ruby on Rails, Daniel Kehoe, Rails Apps Publisher		
Reference Books: 1. Ruby on rails tutorials, Micheal Hartl, Covers Rail Publisher 2. Beginning Ruby, Peter Cooper, Apress.		

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	3	1	-	-	-	-	1	-	-	1
CO2	3	2	3	1	-	-	-	-	1	-	-	-
CO3	3	2	3	1	-	-	-	-	1	-	-	-
CO4	3	2	3	1	1	-	-	-	1	-	-	-
CO5	3	2	3	1	1	-	-	-	1	-	-	1

PEC- CA-301D: Ruby Lab		
Teaching Scheme **Practical: 02 Hours/Week**		**Examination Scheme and Marks** **Internal: 40 Marks** **External: 60 Marks**

Companion Course: Ruby

Course Objectives:

- Understand platform requirements for a robust development environment
- Connect to a Rails application and utilize Rails tools for improved productivity.
- Grasp the principles of the MVC architecture and apply the "Convention over Configuration" philosophy for efficient coding.
- Develop proficiency in Ruby by learning about objects, variables, methods, classes, modules, and built-in classes

Course Outcomes:

On completion of the course, learner will be able to–

CO1: On completion of the course, learner will be able to–
CO2: Develop and test programs using the Ruby programming language.
CO3: Develop, test, and deploy basic web applications with Ruby on Rails (RoR).
CO4: Develop, test, and deploy web layout and user models using RoR.
CO5: Create an advanced project using MySQL, Ruby and the Ruby on Rails framework.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on

various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended :- Window, macOS or Linux

Programming tools recommended: - NetBeans

Virtual Laboratory:

- https://guides.rubyonrails.org/getting_started.html

Part I : Ruby Lab

Suggested List of Laboratory Experiments/Assignments

(6 assignments are compulsory)

Sr. No.	Group A
1.	Write a program for Autoloading.
2.	Write a program to generate the following with usage of MVC and You a) Generating a Model b) Database Migrations c) Using a Model to Interact with the Database d) Showing a List of Articles

3.	Write a program to generate the following with usage of CRUDit Where CRUDit Is Due a) Showing a Single Article b) Resourceful Routing c) Creating a New Article i) Using a Form Builder ii) Using Strong Parameters iii) Validations and Displaying Error Messages iv) Finishing Up
4.	Write a program to Updating an Article a) Using Partials to Share View Code b) Finishing Up
5.	Deleting an Article
6.	Write a program for Adding a Second Model for the following a) Generating a Model b) Associating Models c) Adding a Route for Comments d) Generating a Controller
7.	Write a programming for Refactoring a) Rendering Partial Collections b) Rendering a Partial Form c) Using Concerns
8.	Write a program for Deleting Comments.
	Group B (Mini Project) Select any one problem statement)
1.	to build a blog using RoR

2.	Build a simple product catalogue and shopping cart using RoR
3.	Creating an object-oriented quiz game using RoR
4.	Creating a Twitter or Reddit bot using RoR
5.	Create a simple command-line todo list application where users can add, delete, and mark tasks as complete.
6.	Implement a basic calculator that can perform arithmetic operations like addition, subtraction, multiplication, and division.
7.	Build a program that fetches weather data from an API (such as OpenWeatherMap) and displays the current weather for a specified location.
8.	Develop a program that checks if a given word or phrase is a palindrome (reads the same forwards and backwards).
9.	Create a text-based adventure game where players navigate through different scenarios and make choices that affect the outcome.
10.	Develop a text-based version of the classic Hangman game where players guess letters to reveal a hidden word.

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	3	1	-	-	-	-	1	-	-	1
CO2	3	2	3	1	-	-	-	-	1	-	-	-
CO3	3	2	3	1	-	-	-	-	1	-	-	-
CO4	3	2	3	1	1	-	-	-	1	-	-	-
CO5	3	2	3	1	1	-	-	-	1	-	-	1

Dr. D. Y. Patil Vidyapeeth, Dr. D. Y. Patil School of science & Technology
Second Year BCA (2023 Course)
(With effect from Academic Year 2023-24)

SEMESTER 1V

Course Code	Course Type	Course Name	Teaching Scheme			Examination Assessment Scheme				Credit scheme			
			Lecture	Tutorial	Practical	CA	End Sem	Practical	Total	L	T	P	C
BCA-CA-401	Major	Design & Analysis of Algorithms	3	0	4	40	60	100	200	3	0	4	5
BCA-CA-402	Major	System Programming	4	0	0	40	60	-	100	4	0	0	4
BCA-CA-403	Major	Cloud Computing Methodologies	3	0	4	40	60	100	200	3	0	4	5
PCC-CA-401	VA	Organizational Behaviour	1	0	2	20	30	-	50	1	0	2	2
PEC-CA-401	DSE	A: Advanced Server Side Programming B: Software Application Architecture	2	0	4	40	60	100	200	2	0	4	4

		C: Software Project Management											
HSMC-CA-401	AEC	Ability enhancement	2	0	0	50	-	-	50	2	0	0	2
			15	**0**	**14**	**230**	**270**	**300**	**800**	**15**	**0**	**14**	**22**

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year of BCA (2024-25 Course)

BSC-CA-401 : Design and Analysis of Algorithms

Teaching Scheme:	**Credit**	**Examination Scheme:**
TH: 3 Hours/Week	**5**	**Internal (TH): 40 Marks** **External (TH): 60 Marks**

Prerequisite Courses, if any:

- Data Structures & Algorithms
- Discrete Mathematics

Course Objectives:

- To develop problem solving abilities using mathematical theories.
- To apply algorithmic strategies while solving problems.
- To analyze performance of different algorithmic strategies in terms of time and space.
- To develop time and space efficient algorithms.

Course Outcomes:

On completion of the course, learner will be able to–

CO11: Calculate computational complexity using asymptotic notations for various algorithms.
CO12: Apply Divide & Conquer as well as Greedy approach to design algorithms.
CO13: Understand and analyze optimization problems using dynamic programming.
CO14: Illustrate different problems using Backtracking.
CO15: Compare different methods of Branch and Bound strategy.
CO16: Classify P, NP, NP-complete, NP-Hard problems.

Course Contents

Unit I	**Introduction**	**(06 Hours)**
Algorithm: The Role of Algorithms in Computing - What are algorithms, Design of Algorithm, Analysis of Algorithm: Efficiency- Analysis framework, asymptotic notations – big O, theta and omega. Analysis of Non-recursive and recursive algorithms: Solving Recurrence Equations using Masters theorem and Substitution method. Brute Force method: Introduction to Brute Force method & Exhaustive search, Brute Force solution to 8 queens' problem.		
#Exemplar/Case Studies	Implement Tower of Hanoi	
Mapping of Course Outcomes for Unit I	CO1	
Unit II	**Computational Complexity**	**(06 Hours)**
Non Deterministic algorithms, The classes: P, NP, NP Complete, NP Hard, Satisfiability problem, Proofs for NP Complete Problems: Clique, Vertex Cover		
#Exemplar/Case Studies	Analysis of iterative and recursive algorithm	
Mapping of Course Outcomes for Unit II	CO1, CO6	
Unit III	**Divide & Conquer and Greedy Method**	**(07 Hours)**
Divide & Conquer: Overview, Quick Sort, Binary search, Finding Max-Min, Large integer Multiplication. Greedy Method: General method and characteristics, Kruskal's method for MST, Dijkstra's Algorithm, Fractional Knapsack problem, Job Sequencing, Max flow problem.		
#Exemplar/Case Studies	Study and analyze Merge sort implementation by using Divide and Conquer	
Mapping of Course Outcomes for Unit III	CO1, CO2	
Unit IV	**Dynamic Programming**	**(07 Hours)**

Dynamic Programming: Principle, control abstraction, time analysis of control abstraction, binomial coefficients, Travelling Salesman Problem, OBST, 0/1 knapsack, Chain Matrix multiplication.		
#Exemplar/Case Studies	Study and analyze Fibonacci sequence by using Dynamic Programming.	
Mapping of Course Outcomes for Unit IV	CO1,CO3	
Unit V	**Backtracking and Branch-n-Bound**	**(07 Hours)**
Backtracking: Principle, control abstraction, time analysis of control abstraction, 8-queen problem, graph coloring problem, sum of subsets problem. Branch-n-Bound: Principle, control abstraction, time analysis of control abstraction, strategies- FIFO, LIFO and LC approaches, TSP, knapsack problem.		
#Exemplar/Case Studies	Study of Airline Crew Scheduling	
Mapping of Course Outcomes for Unit V	CO1, CO4, CO5	
Unit VI	**Amortized Analysis**	**(07 Hours)**
Amortized Analysis: Aggregate Analysis, Accounting Method, Potential Function method, Amortized analysis-binary counter, stack Time-Space tradeoff, Introduction to Tractable and Non tractable Problems, Introduction to Randomized and Approximate algorithms, Embedded Algorithms: Embedded system scheduling (power optimized scheduling algorithm), sorting algorithm for embedded systems.		
#Exemplar/Case Studies	Study and analyze cutting stock problem	
Mapping of Course Outcomes for Unit VI	CO3, CO5	
Learning Resources		

Text Books:

5. Parag Himanshu Dave, Himanshu Bhalchandra Dave, “Design And Analysis of Algorithms”, Pearson Education, ISBN 81-7758-595-9
6. Gilles Brassard, Paul Bratley, “Fundamentals of Algorithmics”, PHI, ISBN 978-81-203-1131-2

Reference Books:

11. Michael T. Goodrich, Roberto Tamassia, "Algorithm Design: Foundations," Analysis and Internet Examples‖, Wiley, ISBN 978-81-265-0986-7
12. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to Algorithms", MIT Press; ISBN 978-0-262-03384-8
13. Horowitz and Sahani, "Fundamentals of Computer Algorithms", University Press, ISBN: 978 81 7371 6126, 81 7371 61262
14. Rajeev Motwani and Prabhakar Raghavan, "Randomized Algorithms" Cambridge University Press, ISBN: 978-0-521-61390-3
15. Dan Gusfield, "Algorithms on Strings, Trees and Sequences", Cambridge University Press,ISBN:0- 521-67035-7

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	-	-	-	-	-	-	1	-	-	-
CO2	2	3	-	-	-	-	-	-	1	-	-	-
CO3	2	3	2	-	-	-	-	-	1	-	-	-
CO4	2	3	3	2	-	-	-	-	1	-	-	-
CO5	2	2	2	2	-	-	-	-	1	-	-	-
CO6	2	2	1	2	1-	-	-	-	1	-	-	-

BSC-CA-401 Design and Analysis of Algorithms Lab

Teaching Scheme **Practical: 04 Hours/Week**	**Credit Scheme** **05**	**Examination Scheme and Marks** **Internal: 40 Marks** **External: 60 Marks**

Companion Course: Design and Analysis of Algorithms

Course Objectives:

- To develop problem solving abilities using mathematical theories.
- To apply algorithmic strategies while solving problems.
- To analyze performance of different algorithmic strategies in terms of time and space.
- To develop time and space efficient algorithms.

Course Outcomes:

On completion of the course, learner will be able to–

CO17: Apply and demonstrate Divide & Conquer as well as Greedy approach to design algorithms.
CO18: Apply and analyze optimization problems using dynamic programming.

CO19: Illustrate different problems using Backtracking.
CO20: Demonstrate problems using Branch and Bound strategy.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended :- Windows / Linux

Programming tools recommended: - Turbo C++, Open Source C++ Programming tool like G++/GCC

Virtual Laboratory:

- https://ds1-iiith.vlabs.ac.in/Introduction.html
- https://ds2-iiith.vlabs.ac.in/List%20of%20experiments.html

Part I : Name of the Lab

Suggested List of Laboratory Experiments/Assignments

(6 assignments are compulsory)

Sr. No.	Group A
34.	Using Divide and Conquer Strategies design a function for Binary Search.
35.	Implement Travelling Salesman problem by using Greedy Strategy.
36.	Write a program to implement Min-Max algorithm.
37.	Implement Dijkstras shortest path algorithm by using Greedy Strategy.
38.	Write a program to implement OBST by using Dynamic Programming.
39.	Write a program to implement graph coloring problem by using Backtracking.
40.	Implement 8 Queens problem by using Backtracking.
41.	Implement 0-1 knapsack problem using branch and bound approach
	Group B (Mini Project) **Select any one problem statement**
3	Implement Tower of Hanoi.
3	Implement Chessboard Game..
3	Stochastic Control by using Dynamic Programming.
3	Crossword Puzzle.
3	Implement job scheduling.
3	Implement Sudoku.
3	Build Maze for shortest path.

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	2	2	-	-	-	-	-	-	-	-
CO2	2	3	2	2	-	-	-	-	-	-	-	-
CO3	2	3	3	2	-	-	-	-	-	-	-	-
CO4	2	2	2	2	-	-	-	-	-	-	-	-

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

BCA Second Year (2024-25 Course)

BCA-CA-402: System Programming

Teaching Scheme:	Credit	Examination Scheme:
TH: 4 Hours/Week	**4**	**Internal (TH): 40 Marks** **External (TH): 60 Marks**

Prerequisite Courses, if any:

- Data Structures
- Computer Organization
- C programming

Companion Course, if any: Compiler Design, Operating System

Course Objectives:

- To understand the concepts and components of Systems Programming
- To Learn and understand the fundamentals of Operating systems
- To study the operations performed by Operating System as a resource manager.

Course Outcomes:

On completion of the course, learner will be able to–
CO1: To understand the phases of the compilation process and implementation approach of each phase.
CO2: To understand fundamental principles in compiler design and to provide the skills needed for building compilers.
CO3: To introduce the major concept areas of language translation and compiler design.
CO4: To extend the knowledge of parser by parsing LL parser and LR parser.
CO5: To provide practical programming skills necessary for constructing a compiler.
CO6: To introduce students to the concepts underlying the design and implementation of language processors.

Course Contents

Unit I	**System Software**	**(07 Hours)**
1 Overview of System Software Introduction, Software, Software Hierarchy, Systems Programming, Machine Structure, Interfaces, Address Space, Computer Languages, Tools, Life Cycle of a Source Program, Different Views on the Meaning of a Program, System Software Development, Recent Trends in Software Development, Levels of System Software		
#Exemplar/Case Studies	**Case Study:** Role of system software in OS functionality. **Example:** Exploring kernel operations in Linux for managing hardware resources and providing system services.	
Mapping of Course Outcomes for Unit I	CO1	
Unit II	**Language Processors**	**(08 Hours)**
2 Overview of Language Processors Programming Languages and Language Processors, Language Processing Activities, Program Execution, Fundamental of Language Processing, Symbol Tables Data Structures for Language Processing: Search Data structures, Allocation Data Structures.		
#Exemplar/Case Studies	**Case Study:** Understanding language processors in software development. **Example:** Implementing a simple interpreter for a custom scripting language to execute code directly.	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**Assemblers**	**(07 Hours)**

3 Assemblers Elements of Assembly Language Programming, Design of the Assembler, Assembler Design Criteria, Types of Assemblers, Two-Pass Assemblers, One-Pass Assemblers, Single pass Assembler for Intel x86 , Algorithm of Single Pass Assembler, Multi-Pass Assemblers, Advanced Assembly Process, Variants of Assemblers Design of two pass assembler.		
#Exemplar/Case Studies	**Case Study:** Converting assembly language to machine code. **Example:** Using an assembler to translate ARM assembly instructions into binary code for embedded systems.	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**Macro Processors**	**(08 Hours)**
4. Macro and Macro Processors Introduction, Macro Definition and Call, Macro Expansion, Nested Macro Calls, Advanced Macro Facilities, Design Of a Macro Preprocessor, Design of a Macro Assembler, Functions of a Macro Processor, Basic Tasks of a Macro Processor, Design Issues of Macro Processors, Features, Macro Processor Design Options, Two-Pass Macro Processors, One-Pass Macro Processors.		
#Exemplar/Case Studies	**Case Study:** Automating repetitive tasks in code with macros. **Example:** Creating macros in C to simplify complex code patterns and enhance code reusability.	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Linkers and Loaders**	**(06 Hours)**
5 Linkers and Loaders, Scanning and Parsing Programming Language Grammars, Classification of Grammar, Ambiguity in Grammatic Specification, Scanning, Parsing, Top Down Parsing, Bottom up Parsing, Language Processor Development Tools, LEX, YACC.		
#Exemplar/Case Studies	**Case Study:** Managing program execution with linkers and loaders. **Example:** Using a linker to combine object files and a loader to load the executable into memory.	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**Compiler**	**(06 Hours)**
6 Compilers Causes of Large Semantic Gap, Binding and Binding Times, Data Structure used in Compiling, Scope Rules, Memory Allocation, Compilation of Expression, Compilation of Control Structure, Code Optimization 04 5% 8 Interpreters & Debuggers Benefits of Interpretation, Overview of Interpretation, The Java Language Environment, Java Virtual Machine, Types of Errors, Debugging Procedures, Classification of Debuggers, Dynamic/Interactive Debugger.		

#Exemplar/Case Studies	**Case Study:** Transforming high-level code into machine code. **Example:** Building a simple compiler for a custom programming language to generate efficient executable code.
Mapping of Course Outcomes for Unit VI	CO6.

Learning Resources

Text Books:

7. Aho, A., Lam, M., Sethi, R., & Ullman, J. D. (2006). Compilers: Principles, Techniques, and Tools. 2nd edition. Addison Wesley
8. Chattopadhyaya, S. (2011). System Software. P H I Learning

Reference Books:

1) System Programming by D M Dhamdhere McGraw Hill Publication

2) System Programming by Srimanta Pal OXFORD Publication

3) System Programming and Compiler Construction by R.K. Maurya & A. Godbole.

4) System Software – An Introduction to Systems Programming by Leland L. Beck, 3rd Edition, Pearson Education Asia, 2000

5) System Software by Santanu Chattopadhyay, Prentice-Hall India,2007

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	3	1	-	-	-	-	1	-	-	1
CO2	3	2	3	1	-	-	-	-	1	-	-	-
CO3	3	2	3	1	-	-	-	-	1	-	-	-
CO4	3	2	3	1	1	-	-	-	1	-	-	-
CO5	3	2	3	1	1	-	-	-	1	-	-	1
CO6	3	2	3	1	1	-	-	-	1	-	-	1

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year of Bachelor of Computer Application (2024-25 Course)

BCA-CA-403 : Cloud Computing Methodologies

Teaching Scheme:	**Credit**	**Examination Scheme:**
TH: 3 Hours/Week	5	**Internal (TH): 40 Marks** **External (TH): 60 Marks**

Prerequisite Courses, if any:

- Fundamentals of Embedded Systems, IoT
- Basic of Computer Networking, data communication ,

Companion Course, if any: Embedded Systems and IoT

Course Objectives:

1. To introduce the fundamentals of cloud computing, its technologies, Challenges and Applications
2. To give Insights into the virtualization technologies and Architecture
3. To know the relationship between Cloud and SOA
4. To classify and evaluate Cloud Security Issues
5. To apply theory to practical knowledge through case Studies

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Describe the concepts of Cloud Computing and its ServicModels&Deployment Models.
CO2: Classify the types of Virtualization.
CO3: Describe the Cloud Management and relate Cloud to SOA.
CO4: Interpret Architecture and Pharrell Programing of Cloud Computing.
CO5: Demonstrate practical implementation of Cloud computing.
CO6: Design and implement security measures to protect cloud computing environments from potential threats and vulnerabilities.

Course Contents

Unit I	**Cloud Services and Cloud Models**	**(07 Hours)**

Cloud Services and Cloud Models- Introduction to Cloud,Cloud Computing vs. Cluster Computing vs. GridComputing,Introduction to Cloud Service Models,Characteristics, Advantages, Security,XAAS- Anything as a Service – Storage as a service,Network as a Service, Database as a Service etc.,IAAS, PAAS, SAAS characteristics, benefits andApplications,Comparison of SAAS, PASS and IAAS,Cloud Deployment Models-Public, Private, Hybrid,Cloud Platforms :Google Cloud Platform,Microsoft Azure,SalesForce,AWS.

#Exemplar/Case Studies	Cloud Computing for Government
Mapping of Course Outcomes for Unit I	CO1

Unit II	**Virtualization**	**(08 Hours)**

Virtualization--Introduction to Virtualization concept &Hypervisors,Types of Virtualization: Server, Storage and Network,Pros and Cons of Virtualization,Machine Image, Virtual Machine (VM),Technology Examples,Xen: Para virtualization,VMware: Full Virtualization,Open Source Virtualization Manager.

#Exemplar/Case Studies	Evaluate the features and performance of Xen hypervisor compared to other virtualization solutions.	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**SOA & Cloud Management**	**(08 Hours)**
SOA & Cloud Management- Definition of Service Oriented Architecture,Basic concepts of SOA,Web Services: SOAP and REST,Cloud APIs (RESTful),Relating SOA and Cloud Computing.,Cloud Availability,Cloud Governance,Service Level Agreement		
#Exemplar/Case Studies	Pricing Model: Usage Reporting, billing andmetering (AWS), Cloud Statistics	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**Multi Core Architecture**	**(08 Hours)**
Multi Core Architecture- Cloud Computing Architecture,Multi Core Architecture,Multi Cloud Environment,Parallel Programming,Parallel Processing,Edge Computing Concepts		
#Exemplar/Case Studies	Design and implement a multi-core architecture for edge computing deployment.	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Moving Applications to the Cloud**	**(06 Hours)**
Moving Applications to the-Cloud Migration Strategies and Process,Issues in Inter Cloud,Applications in the Clouds,Cloud Service Attributes,Cloud Bursting,Data Migration in Cloud,Quality of Services in cloud Computing		
#Exemplar/Case Studies	Six R for Cloud Migration	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**Cloud Security & Implementation of Cloud**	**(06 Hours)**

Cloud Security & Implementation of Cloud- Cloud Security Fundamentals,Cloud Security Architecture ,Cloud Computing Security Challenges,Privacy and Security in Cloud,Identity Management and Access control,Demonstrate the commercial cloud computingInfrastructures,Introduction to Dockers Container	
#Exemplar/Case Studies	Design a comprehensive cloud security architecture tailored to the organization's requirements.
Mapping of Course Outcomes for Unit VI	CO6

Learning Resources

Text Books:

9. Cloud Computing Black Book, Kailash Jayaswal (Author), Jagannath Kallakurchi (Author), Donald J. Houde (Author), Dr. Deven Shah (Author), ISBN: 978-9351193944.

10. Cloud Computing For Dummies by Judith Hurwitz, Robin Bloor, Marcia Kaufman, and Fern Halper ISBN: 978-1119546658

Reference Books:

1. Cloud Computing Bible by Barrie Sosinsky, Wiley India Pvt. Ltd,
2. Cloud Computing : Automating the Virtualized Data Center
3. Cloud Computing by Dr. Kumar Saurabh ,Wiley–India
4. Cloud computing: A practical approach by Anthony T. Velte, TataMcGraw-Hill
5. Cloud Computing Concepts, Technology & Architecture by Thomas Erl,Zaigham Mahmood, and Ricardo Puttin
6. Mastering Cloud Computing by Rajkumar Buyya, Christian Vecchiola,S.Thamarai Selvi - McGraw Hill Education (India) Private Limited,
7. Cloud Computing Web –Based Applications that change the way youwork and Collaborate Online by Michael Miller, Pearson
8. Cloud Computing for Dummies by Judith Hurwitz, Robin Bloor, MarciaKaufman, FernHalper

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	2	1	2	3	-	-	1	-	-	1
CO2	3	2	3	1	3	-	-	-	1	-	-	-
CO3	3	2	3	1	3	-	-	-	1	-	-	-
CO4	3	2	3	1	1	-	-	-	1	-	-	-

CO5	3	2	3	1	1	-	-	-	1	-	-	1
CO6	3	2	3	1	1	-	-	-	1	-	-	1

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year of Bachelor of Computer Application (2024-25 Course)

Cloud Computing Methodologies

Teaching Scheme **Practical: 4 Hours/Week**	**Credit Scheme** **05**	**Examination Scheme and Marks** **Internal: 40 Marks** **External: 60 Marks**

Companion Course: ESC-CS 601: Cloud Computing

Course Objectives:

- To learn basic concepts ,types and characteristics of cloud computing
- To learn Cloud Computing Architecture and service models.
- To learn Virtualization and its type's in cloud computing.
- To learn fundamental concepts and architecture of cloud computing security.
- To learn basics of SOA and cloud based storage

Course Outcomes:

On completion of the course, learner will be able to–

CO1 : Able to understand basic concepts, principles and paradigm of Cloud Computing
CO2 : Understand the various Cloud computing models and services.
CO3 : Able to identify the significance of implementing virtualization techniques.
CO4 : Able to understand the need of security in Cloud computing.
CO5 : Understand the concept SOA and cloud based storage in Cloud computing.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended :- Ghost OS,CloudMe

Programming tools recommended: - CloudZero,Amazon Web Services,Google App Engine

Virtual Laboratory:

- https://vlab.noaa.gov/web/osti-modeling/cloud-computing1
- https://www.codio.com/solutions/virtual-labs

Part I : Cloud Computing	
Suggested List of Laboratory Experiments/Assignments **(6 assignments are compulsory)**	
Sr. No.	**Group A**
42.	Create an Account to Cloud Service Provider (AWS, AZURE, Google Cloud, etc.)
43.	Create an Instance on Cloud
44.	Provide Access Control and Permission to Users
45.	Execute the Web Page on Cloud
46.	Provide Security Mechanism to your instance.
47.	Create an Account to Cloud Service Provider (AWS, AZURE, Google Cloud, etc.)
48.	Create an Instance on Cloud
49.	
	Group B (Mini Project) **Select any one problem statement**
3	E-learning Platform
3	**Information Chatbot**
3	**Secure File Storage System**
4	**Smart Traffic Management Solution**
4	**Movie Recommendations Application**
4	**Bus Ticketing System with Payment Capabilities**
4	**Virtual Event Management Platform**

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	1	1	1	1	-	1	1	-	-	1

CO2	1	2	1	3	1	-	1	-	1	-	-	-
CO3	-	2	3	1	2	1	-	1	1	-	-	-
CO4	2	1	2	1	-	2	1	-	1	-	-	-
CO5	1	1	1	2	1	2	-	1	1	-	-	1

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year of Bachelor of Computer Applications (2024-25 Course)

PCC-CA-401: Organizational Behavior

Teaching Scheme:	**Credit**	**Examination Scheme:**
TH: 01 Hours/Week **PR: 02 Hours/Week**	**02**	**Internal (TH): 20 Marks** **External(TH): 30 Marks**

Prerequisite Courses, if any:

- Students must have knowledge communication skills and human values

Companion Course, if any: -

Course Objectives:

- To study the fundamental concepts of Organization Behavior.
- To understand the impact of individual and group behavior on organizational effectiveness.
- To learn on the motivation and leadership influence to Behavior and Performance.
- To learn on Group Dynamics of people management and conflict management.
- To understand the diverse work culture and essence of Quality Work Life in an Organization.

Course Outcomes:

On completion of the course, learner will be able to–
CO1: Analyze Organizational Behavior and Management along with the Basic Behavioral Science that influence Organizational Behavior.
CO2: recognizing and valuing individual Personalities and Behaviors by working on Perceptions from Organizational Perspective
CO3: Develop good Work Culture and Climate in an Organization
CO4: recognize good and bad leadership for the organization
CO5: Analyze the influence of Individual and Group Behavior towards meeting the Organizational Goal.
CO6: resolve conflict at the interest of the common Organizational Goal.

Course Contents

Unit I	**Introduction to Organizational Behaviour**	**(03 Hours)**
Management and Organizational Behavior, Theories of Management, Major Behavioral Science that contribute to Organizational Behavior-Psychology, Sociology, Socio-Psychology, Political Science, Anthropology, Organizational structure, Dynamics of People and Organization, Models of Organizational Behavior, Hawthorne studies, Challenges and opportunities in Organizational Behavior		
#Exemplar/Case Studies	Coca Cola Case Study: An Analysis of Organizational Behavior	
Mapping of Course Outcomes for Unit I	CO1	
Unit II	**Motivation, Personality & Perception**	**(05 Hours)**
Motivation-Motivation and Behavior, theories of Motivation, Reinforcement theory, Organizational Learning Process, Motivation and performance, Financial and Non-financial incentives, Personality Determinants of personality, Type A and Type B personality, Values, Attitudes & Beliefs, Argyris'sMaturity-Immmaturity Continuum, Perception-Motivation and Perception, Meaning, Need of Perceptual process, Factors influencing Perceptual process, self-concept and self-esteem.		
#Exemplar/Case Studies	Netflix Inc.'s Organizational Structure and its strategic Implications	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**Group Dynamics and Stress Management:**	**(04 Hours)**

Group Dynamics-Team & Group difference, Group Effectiveness, Formal & Informal Group, Stages of Group Development, Group Decision Making, Inter group relation and Conflict, Stress Management-Stress and Behavior, Sources of Stress, Consequences of Stress and Performance		
#Exemplar/Case Studies	Bella's: a case study in organizational behavior	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**Leadership**	**(03 Hours)**
Leadership-Introduction and characteristics of Leadership, Formal and Informal leadership, Theories of Leadership, Leadership Qualities, Leadership vs. management, Leadership styles.		
#Exemplar/Case Studies	A Case Study on Women leadership in Panchayati Raj Institutions (PRI) at the Gram Panchayat level	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Conflict Management and Power & Politics**	**(05 Hours)**
Conflict Management-Nature of Conflict, Sources of Organizational Conflict, Modes of Conflict Resolution, Conflict Management, Power & Politics-Difference between Influence, Power &Authority, Sources of power, Organizational Politics, Machiavellian ism, Ethics of Power and Politics in Organizations.		
#Exemplar/Case Studies	A case study on Maruti Suzuki Ltd	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**Organization Development and Culture**	**(04 Hours)**
Organizational Change, Resistance to change, Steps for planned change, Quality Work Life, Organization Development Objective and Interventions, Organization Climate and Organizational Effectiveness, Managing Organizational Culture		
#Exemplar/Case Studies	Organizational Socialization in Professional Sport: The National Basketball Association's Rookie Transition Program	
Mapping of Course Outcomes for Unit VI	CO6	
Learning Resources		

TEXT BOOKS:

1. Uma Sekaran, Organisational Behaviour, Tata McGraw Hill
2. John W Newstrom, Organisational Behaviour, Tata McGraw Hill

Reference Books:

1. Stephen P.Robbins, Timothy A. Judge, Niharika Vohra (18th ed.), Pearson Education, New Delhi4. L. M. Prasad, Organisational Behaviour, Sultan Chand & Sons

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	3	-	-	-	-	-	-	-	-	-
CO2	1	1	-	-	2	-	-	-	1	-	-	1
CO3	2	-	1	-	1	-	-	-	2	-	2	-
CO4	-	-	2	-	-	-	-	-	-	-	2	-
CO5	2	1	-	-	2	-	-	-	-	-	-	1
CO6	2	1	-	-	-	-	-	-	-	-	-	-

PCC-CA-401: Organizational Behavior

Teaching Scheme **Practical: 02 Hours/Week**		**Examination Scheme and Marks** **Internal (PR): 20 Marks** **External (PR): 30 Marks**

Companion Course: PEC-CA-401: Software Project Management

Course Objectives:

- Understand the various stages of testing
- Appreciate the use of tools for verification and validation
- Appreciate the benefits of using metrics for verification and validation

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Design Architecture of given system.
CO2: Create basic UML diagrams for real world application

CO3: Employ various software architecture design components.
CO4: Design methods for improving software quality from the perspective of software architecture.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended: - Windows / Linux

Programming tools recommended: - SysML/ StarUML, Selenium

Part I : Software Application Architecture

Suggested List of Laboratory Experiments/Assignments

(6 assignments are compulsory)

Sr. No.	Group A
50.	Let us examine the problem faced by Mr. Nataraj, Regional Manager of Alpha Pvt. Ltd. Alpha makes and distributes products from more than 10international pharmaceutical and health care companies. Mr. Nataraj is responsible for managing existing clients and also to get new clients. He manages a number of sales representatives. Important customers have a dedicated sales representatives, while other sales representatives try to get new clients. One day an important customer (Good Health Hospital) called Mr. Nataraj and complained that Mr. Bhavan (the sales representative) was ineffective and insisted he be removed, or else they would not give any business. Here are Mr. Nataraj's thoughts: • The track record of Mr. Bhavan was good and he was liked within the company. Dismissing him or even transferring him to a new region will affect the morale of the work force. •Good health hospitals is a major customer and gives good business. Loosing the hospital is not an option. Therefore the demands of the hospital have to be met. **Q . If You were Mr. Natraj, how will you solve this issue?**

51.	Krishnamurthy, plant manager of frame manufacturing company, is the chairperson of the ad hoc committee for space utilization. The committee is made up of various departmental heads of the company. The general manager of the company has given MURTHY the responsibility for seeing whether the various office, operations and warehouse facilities of the company are being optimally utilized. The company is beset by rising costs and the need for more space. However, before Okaying an expensive addition to the plant, the general manager wants to be sure that the currently available space is being utilized properly MURTHY opened up the first committee meeting byReiterating the charge of the committee. Then MURTHY asked the members if they had any initial observations to make. The first to speak was the office manager. He stated "well IKnow we are using every possible inch of room that we have available to us. But when I walk out into the plant I see lot of open spaces. We have people piled on top of oneAnother, but out in the plant there seems to be plenty of room." the production manager quickly replied, "We do not have a lot of space. You office people have the luxury facilities. My supervisors don't even have room for descend a file cabinet. i have repeatedly told the plant manager we need more space. After all, our operation determines whether this plant succeeds or fails, not like you people inThe front office pushing paper around.' MURTHY interrupted at this point and said, "Obviously we have different interpretations of the space utilization around here. BeforeFurther discussion I think it would be best if we have some objective facts to work with. I am going to ask the industrial engineer to provide us with some statistics on plant and office layouts before our next meeting. Today's meeting is adjourned **QUESTIONS: WHAT PERCEPTUAL PRINCIPLES ARE EVIDENT IN THIS CASE ?**
52.	One Monday morning Sanjay Nagpal, a recent recruit from a reputed anagement institute in Chennai walked into the sales office at maniple as a new sales trainee. Raghavan the zonal sales manager for a large computer hardware firm was there to greet him. Raghavan's job consisted of overseeing the work of sales officers, field executives and trainee salesman numbering over 50 of three areas namely manipal, Bangalore. Trivandrum. The sales growth of computers, parts and other office equipments in his area was highly satisfactory, especially in recent years – thanks to the developmental initiatives taken by respective state government in spreading computer education in office, schools, college, banks and other institutions. Raghavan had collected several sales reports, catalogues, and pamphlets describing in detail the types of office equipment sold by the company. After a pleasure chat about their backgrounds, Raghavan gave sanjay the collected material and showed him to his assigned desk. Thereafter Rahavan excused himself and did not return. Sanjay spent the whole day scanning the material and at 5.00 P.m. he picked up his things and went home. **Q: what do you think about Raghavan's approach for training Programme?**
53.	Prepare a case study report on Leadership in Infosys Technologies

54.	Organizational Socialization in Professional Sport: The National Basketball Association's Rookie Transition Program
55.	A case study on Maruti Suzuki Ltd
56.	Any other case studies
	Group B (Mini Project) **Select any one problem statement**
4	A case study of Organizational Behaviour and Resistance to changes in Malaysia's Commercial Banking Industry
4	Assessing Organizational Behavior: A Case Study in a Colombian Retail Store
4	Case Study - A Turnaround at Tentex
4	Case Study - The Key-Man Syndrome
4	Case Study - Engine Solutions (ES) Acquires JNC
4	Managing the Emotional Employee in the Work Setting
5	Any other (choice of student)

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2		2	3	-	-	-	-	-	-	1
CO2	1		2	2	2	-	-	-	-	-	-	1
CO3	1	2	2	2	2	-	-	-	-	-	-	1
CO4	2	1		1	1							

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year of BCA (Bachelor of Computer Application)

(2024-25 Course)

PEC- CA-401A: Discipline Specific Elective -4

(Advanced Server-Side Programming)

Teaching Scheme:	Credit	Examination Scheme:
TH: 2 Hours/Week	04	Internal (TH): 40 Marks External (TH): 60 Marks

Prerequisite Courses, if any: HTML, CSS

Companion Course, if any: **PHP**

Course Objectives:

- Understand how server-side programming works on the web.
- Using PHP built-in functions and creating custom functions
- Understanding POST and GET in form submission.
- How to receive and process form submission data.
- Read and process data in a MySQL database.

Course Outcomes:

On completion of the course, learner will be able to–

CO1: To understand basics of PHP, HTML and implement PHP script using Decisions and Loops.

CO2: To implement PHP script using Strings and Functions.

CO3: To develop PHP applications using Arrays and Functions.

CO4: Implements opening, reading and Writing of Files.

CO5: To develop the connection between PHP and MySQL

CO6: To understand controlling session and setting cookies.

Course Contents

Unit I	**Introduction to HTML, HTTP and PHP**	**(07 Hours)**

Overview of HTML and Basic Tags , Creating Forms ,Tables, HTML5 Semantics. CSS basic concept ,Three ways to use CSS, Box Model, Navigation Bar . Introduction to Web server and Web browser . HTTP basics . PHP Basics: Use of PHP, Lexical structure, Language basics, control structures- looping and branching statements, decision making statements.		
#Exemplar/Case Studies	Create the web application using html and css that contain tables, forms and lists	
Mapping of Course Outcomes for Unit I	CO1	
Unit II	**Function and String**	**(08 Hours)**
Defining and calling a function, Default parameters, Variable parameters, Missing parameters, Variable function, Anonymous function, Types of strings in PHP, Printing functions, Encoding and escaping, Comparing strings, Manipulating and searching strings, Regular expressions		
#Exemplar/Case Studies	Developing Content Management System with Enhanced String Handling	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**Arrays**	**(07 Hours)**
Indexed Vs Associative arrays, Identifying elements of an array, Storing data in arrays, Multidimensional arrays, Extracting multiple values- Converting between arrays and variables, Traversing arrays, Sorting, Action on entire array		
#Exemplar/Case Studies	Create a web application that sorting the array elements and traversing the array.	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**Files**	**(07 Hours)**
Working with files and directories, Opening and Closing, Getting information about file, Read/write to file, Splitting name and path from file, Rename and delete files, Reading and writing characters in file, Reading entire file.		
#Exemplar/Case Studies	Create a web application for opening, reading and Writing of Files.	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Database Connectivity Using PHP**	**(07 Hours)**

Introduction to MySQL - Data types, attributes, working with databases, working with tables, altering table structure; Database Connectivity-Using the MYSQL extension, setting up the connection, handling errors, querying the database, working with prepared statements, auto commit mode, committing and rolling back a transaction.		
#Exemplar/Case Studies	Develop a web application that creates a database and connects to the web page.	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**Session and Cookie**	**(06 Hours)**
Introduction to Session Control, Session Functionality What is a Cookie, Setting Cookies with PHP. Using Cookies with Sessions, Deleting Cookies, Registering Session variables, Destroying the variables and Session.		
#Exemplar/Case Studies	Enhancing User Experience through Session and Cookie Management in a Web Application	
Mapping of Course Outcomes for Unit IV	CO6	

Learning Resources

Text Books:

1. Beginning PHP, Apache, MySQL web development

Reference Books:

1. HTML & CSS: The Complete Reference, Fifth Edition Author: Thomas A. Powell First published: 01 Jan 2010.
2. Programming PHP By Rasmus Lerdorf and Kevin Tatroe, O'Reilly publication
3. Beginning PHP 5 , Wrox publication
4. PHP web sevices, Wrox publication
5. Mastering PHP , BPB Publication
6. PHP cookbook, O'Reilly publication
7. PHP for Beginners, SPD publication
8. Programming the World Wide Web , Robert W Sebesta(3rd Edition)
9. HTML 5 Black Book : Covers Css3, Javascript, XML, XHTML, Ajax, PHP And Jquery by Kogent Learning Solutions Inc, Published November 2011 by Dreamtech Press
10. Spurlock Jake, Bootstrap: Responsive Web development. O'Reilly Media, Inc

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	-	-	-	-	-	-	-	-	-	2
CO2	2	1	2	-	-	-	1	1	2	-	1	-
CO3	1	2	3	1	2	1	1	1	2	-	1	-
CO4	1	2	2	-	2	1	-	-	-	-	1	-
CO5	3	1	3	-	-	-	1	-	-	1	-	1

PEC- CA-401A: Discipline Specific Elective -4 Lab

(Advanced Server-Side Programming)

Teaching Scheme		**Examination Scheme and Marks**
Practical: 04 Hours/Week		**Internal: 40 Marks** **External: 60 Marks**

Companion Course: C#

Course Objectives:

- Gain the PHP programming skills needed to successfully build interactive, data-driven sites.

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Develop programs involving basic features of PHP programming.

CO2: Develop Array and function.

CO3: Make use of Session and cookies features to safeguard program against runtime anomalies.

CO4: To Develop interactive websites.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended :- Windows

Programming tools recommended: - PHPStorm, XAMPP, MySQL

Virtual Laboratory:

- https://www.vlab.co.in/lab_ready_for_use.php

Part I : PHP Lab

Suggested List of Laboratory Experiments/Assignments (6 assignments are compulsory)	
Sr. No.	**Group A**
1.	Get name of the user from a form and show greeting text.
2.	Write a php program to check whether given number is palindrome or not.
3.	Write a Age calculator program.
4.	Write a php program to Array manipulation.
5.	Write a php program hit counter using cookies.
6.	Create a PHP page for login page without sql connection.
7.	Write a php program to Write a file.
8.	Write a php program to design personal information.
	Group B (Mini Project) Select any one problem statement
1.	Transport Management System
2.	Library Management System
3.	Two-wheeler Rental System.
4.	Complaint Management System.
5.	Vehicle Breakdown Assistance (On-Road)
6.	Gym Management System.
7.	School Security System.
8.	Fake Review Identification.

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	1	1	2	1	-	-	1	1	-	2
CO2	2	2	1	1	2	-	-	-	1	-	1	2
CO3	2	2	2	2	1	1	-	-	2	-	-	2
CO4	2	2	1	1	3	-	-	-	1	1	-	2

CO5	2	2	2	2	2	1	-	-	2	-	-	1

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year of Bachelor of Computer Applications (2024-25 Course)

PEC-CA-401B: Software Application Architecture

Teaching Scheme:	Credit	Examination Scheme:
TH: 02 Hours/Week **PR: 04 Hours/Week**	**04**	**Internal (TH): 40 Marks** **External(TH): 60 Marks**

Prerequisite Courses, if any:

- Fundamentals of programming and object oriented programming concepts

Companion Course, if any: - PEC-CA-401**:** Software Project Management

Course Objectives:

- Understand the fundamentals of software architecture.
- Study the various software development methodologies.
- Learn the importance of architectural documentation and evaluation.
- Learn the various software architecture design components.
- Relate software architecture and software quality.

Course Outcomes:

On completion of the course, learner will be able to–
CO1: Develop Software applications starting from software architecture and design.
CO2: Learn and evaluate existing software architectures.
CO3: Realize importance of architectural documentation and document them
CO4: Evaluate the architectural model.
CO5: Employ various software architecture design components.
CO6: Design methods for improving software quality from the perspective of software architecture.

Course Contents

Unit I	Introduction	(06 Hours)

Basic concepts of software architecture – Context of Software Architecture – ABC cycle – What software architecture is and what it isn't – Architectural patterns – Good Architecture- Reference models – Architectural structures and views-Introduction to styles – Decentralized Architectures.

#Exemplar/Case Studies	College Management system
Mapping of Course Outcomes for Unit I	CO1

Unit II	DESIGN METHODOLOGIES	(05 Hours)

Structured design- Design practices-Stepwise refinement – Incremental design- Structured system analysis and design –Jackson structured programming – Jackson system Development.		
#Exemplar/Case Studies	Library Management system	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**ARCHITECTURAL DESCRIPTION DOCUMENTATION AND EVALUATION**	**(06 Hours)**
Early architecture description languages-Domain and style specific ADL's- Extensible ADL's – Documenting software architecture – Uses and Audiences for Architecture Documentation – Views – Choosing Views – Combining Views –		
#Exemplar/Case Studies	Hospital management system	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**ARCHITECTURAL EVALUATION**	**(05 Hours)**
Architecture evaluation – Evaluation Factors –Architecture Tradeoff Analysis Method – Lightweight Architecture Evaluation – ATAM.		
#Exemplar/Case Studies	Online shopping	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**ARCHITECTURE DESIGN**	**(06 Hours)**
Typical architectural design-Dataflow-Independent components-Call and return – Using styles in design – Architectural design space-Design space of architectural elements – Design space of architectural styles		
#Exemplar/Case Studies	Web application Login Controller	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**IMPLEMENTATION AND CONFORMANCE TO ARCHITECTURE**	**(07 Hours)**

Understanding quality attributes- Implementation of Quality attributes in Architecture – Architecture and requirements conformance –Functionality– Quality attribute considerations – System quality attributes-Introduction to tactics – Achieving Quality Attributes through Tactics –Tactics types – Architectural patterns and styles – Architecture and Quality Attributes – Quality attribute scenarios in practice.	
#Exemplar/Case Studies	Web application Login Controller
Mapping of Course Outcomes for Unit VI	CO6

Learning Resources

TEXT BOOKS:

1. Len Bass, Paul Clements, Rick Kazman, "Software Architecture in Practice", Third Edition, Addison,Wesley, 2012.

2. David Budgen, "Software Design", Second Edition, Pearson Education, 2004.

Reference Books:

16. 1. Richard N.Taylor, NenadMedvidovic and Eric M.Dashofy, "Software Architecture, Foundations,Theory and Practice", Wiley 2010.
17. Hong Zhu, "Software Design Methodology from Principles to Architectural Styles", Elsevier, 2005.
18. Mary Shaw and David Garlan, "Software Architecture –Perspectives on an emerging Discipline",Pearson Education, 2008.

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	3	-	-	-	-	-	-	-	-	-
CO2	1	1	-	-	2	-	-	-	1	-	-	1
CO3	2	-	1	-	1	-	-	-	2	-	2	-
CO4	-	-	2	-	-	-	-	-	-	-	2	-
CO5	2	1	-	-	2	-	-	-	-	-	-	1
CO6	2	1	-	-	-	-	-	-	-	-	-	-

PEC-CA-401B: Software Application Architecture

Teaching Scheme		Examination Scheme and Marks
Practical: 04 Hours/Week		**Internal (PR): 40 Marks** **External(PR): 60 Marks**

Companion Course: PEC-CA-401: Software Project Management

Course Objectives:

- Learn to create architectural documentation and evaluation.
- Apply the various software architecture design components.
- Create software architecture and check software quality.

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Design Architecture of given system.
CO2: Evaluate the architectural model.
CO3: Employ various software architecture design components.
CO4: Design methods for improving software quality from the perspective of software architecture.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended: - Windows / Linux

Programming tools recommended: - Diagrams.net / Excalidraw

Part I : Unified Modeling Language diagrams

Suggested List of Laboratory Experiments/Assignments

(6 assignments are compulsory)

Sr. No.	Group A
57.	Document the Software Requirements Specification (SRS) for the identified system
58.	Apply Architectural patterns on given problem statement.
59.	Create Incremental design for given problem statement
60.	Create the document for created architecture
61.	Evaluate the architectural model
62.	Create architecture of data flow in system
63.	Implement Quality attributes in Architectural model.
	Group B (Mini Project) **Select any one problem statement**
5	e-Library online public access catalog (OPAC)

	5	Restaurant business model
	5	Online shopping system
	5	Hospital Management
	5	Software protection and licensing
	5	Online ticket booking System
	5	Netflix
	5	Any real world application other (choice of student)

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2		2	3	-	-	-	-	-	-	1
CO2	1		2	2	2	-	-	-	-	-	-	1
CO3	1	2	2	2	2	-	-	-	-	-	-	1
CO4	2	1		1	1							

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year BCA(SEM 4)(2024-25 Course)

PEC-CA-401C: Software Project Management

Teaching Scheme:	Credit	Examination Scheme:
TH: 02 Hours/Week	04	**Internal (TH): 40 Marks** **External (TH): 60 Marks**

Prerequisite Courses, if any:

- Students must have a knowledge of fundamentals of software programming

Companion Course, if any: Software Engineering

Course Objectives:

Course Objective:

- To understand the fundamental of software engineering
- To discuss about the requirement analysis and design using various tools.
- To differentiate the role of software developer and software tester.
- To illustrate the use of COCOMO models for projects cost estimation.
- T5. To provide a working knowledge of estimating, design, testing, and quality management strategies for big software development projects.
- To conceptualize the Software Development Life Cycle (SDLC) models.

Course Outcomes:

On completion of the course, learner will be able to–

CO1: To understand the need of software engineering and its various models.

CO2: To interpret the phases of Software Development using agile methodology.

CO3: To classify various Lifecycle models, requirement analysis and specifications.

CO4: To understand preparation of SRS document, Design Concepts.

CO5: To understand and demonstrate software coding, software testing for a given set of problem

CO6: To familiarize Project Management framework and Tools.

Course Contents

Unit I	Introduction	(04 Hours)

Importance and Emergence of Software Engineering Feasibility Study, Requirement Analysis, Design, Implementation, Testing, and Maintenance phases of software development Software Life Cycle Models: Waterfall, Iterative, Prototyping, Spiral, and Agile - Compare and contrast life cycle models.

#Exemplar/Case Studies	Feasibility Study, Requirement Analysis for library management system	
Mapping of Course Outcomes for Unit I	CO1	
Unit II	**Requirements Analysis and Design**	**(04 Hours)**
Process of analysis, Requirement specification, ideal SRS properties, SRS document structure, etc. Diagrams of Data Flow - Software Architecture and Architecture Views: What Role Do They Play? - Software Project Planning Software Design - Software Design Concepts - Complexity Metrics for Function-Oriented Design - Complexity Metrics for Object-Oriented Design - A well-thought-out design. Use Case Approach. Requirement Analysis and Design: DFD, Data Dictionary		
#Exemplar/Case Studies	SRS for library management system	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**Software Project Planning & Management**	**(04 Hours)**
Business Case, Project selection and Approval, Project charter, Project Scope management: Scope definition and Project Scope management, Creating the Work Breakdown Structures, Scope Verification, Scope Control, Methods for estimating project time and cost, Resource Management		
#Exemplar/Case Studies	**The Tesla Electric Car Project**	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**Project Scheduling**	**(04 Hours)**
Relationship between people and Effort: Staffing Level Estimation, Effect of schedule Change on Cost, Degree of Rigor & Task set selector, Project Schedule, Schedule Control, CPM (Numerical), Basic Planning Purchases and Acquisitions, Planning Contracting, Requesting Seller Responses, Selecting Sellers,		
#Exemplar/Case Studies	**The Tesla Electric Car Project**	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Agile Methodology**	**(04 Hours)**
Theories for Agile Management – Agile Software Development – Traditional Model vs. Agile Model - Classification of Agile Methods – Agile Manifesto and Principles – Agile Project Management – Agile Team Interactions – Ethics in Agile Teams		

#Exemplar/Case Studies	**The Tesla Electric Car Project**	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**Business Continuity & Disaster Management**	**(04 Hours)**
Introduction to Disaster Recovery and Business Continuity, Nature and Causes of Disasters, Business Continuity Management, Disaster Recovery Planning Process		
#Exemplar/Case Studies	Air crash investigation	
Mapping of Course Outcomes for Unit VI	CO6	

Learning Resources

Text Books:

1. alker Royce, "**Software Project Management**", Addison-Wesley, 1998

2. **Software Project Management**: Saikat Dutt /S. Chandramouli, Pearson-Second Edition

Reference Books:

1. **Software Engineering**, 5th and 7th edititon, by Roger S Pressman, McGraw Hill publication.

2. **Managing Information Technology Project**, 6edition, by Kathy Schwalbe, Cengage Learning publication. Information Technology Project Management by Jack T Marchewka Wiley India publication.

3. **Software Engineering** 3rd edition by KK Agrawal, Yogesh Singh, New Age International publication. Software Engineering Project Management by Richard H. Thayer Wiley India Publication.

4. **Software Engineering**, A Precise Approach: Pankaj Jalote, Wiley India-2010 2.

5. **Software Project Management**: Saikat Dutt /S. Chandramouli, Pearson-Second Edition

6. Ken Schwaber, "**Agile Project Management**", Microsoft Press, 2004

@The CO-PO mapping table

CO\ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12

CO1	1	1	2	1	-	-	-	-	-	-	-	-
CO2	1	2	-	2	-	-	-	-	-	-	-	-
CO3	2	1	2	1	-	-	-	-	-	-	-	-
CO4	1	2	-	2	-	-	-	-	-	-	-	-
CO5	-	-	2	-	-	-	-	-	-	-	-	-
CO6	-	2	1	2	-	-	-	-	-	-	-	-

PEC-CA-401C: Software Project Management Lab		
Teaching Scheme **Practical: 04 Hours/Week**	**Credit Scheme** **04**	**Examination Scheme and Marks** **Internal: 40 Marks** **External: 60 Marks**
Companion Course: Software Engineering		
Course Objectives: • Apply various software engineering concepts for real world applications. • Apply various project management concepts for real world applications.		
Course Outcomes: On completion of the course, learner will be able to– • **CO1:** Understand real world problem statements. • **CO2**: Create project schedule. • **CO3**: Understand and apply the software testing concepts.		
Guidelines for Instructor's Manual The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..		

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Virtual Laboratory:

-

Part I : Software Project Management

Suggested List of Laboratory Experiments/Assignments

(6 assignments are compulsory)

Sr. No.	Group A

64.	Formulation of a problem statement.
65.	Documentation for the Software Requirement Specification Document, Design Documents, and Documentation for the Testing Phase.
66.	Documentation relating to Software Configuration Management and Risk Management.
67.	Create Project schedule using any tool (e.g. MS Project)
68.	Research and application of any CASE tool for the design phase
69.	Using any Design phase CASE tools to complete the design.
70.	Create unit testing and integration testing test cases.
71.	Create test cases for a variety of white-box and black-box testing methods.
	Group B (Mini Project) **Select any one problem statement**
5	Online hotel booking systems
6	Stock Market Risk Analysis
6	Hospital Management System
6	Shopping Mall Inventory Management
6	Student Attendance Management System
6	Restaurant Management system
6	Railway reservation system

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	2	2	3	-	-	-	-	-	-	1
CO2	1	2	2	2	2	-	-	-	-	-	-	1
CO3	1	2	2	2	2	-	-	-	-	-	-	1

Dr. D. Y. Patil Vidyapeeth, Dr. D. Y. Patil School of science & Technology
Third Year BCA (2023 Course)
(With effect from Academic Year 2023-24)

SEMESTER V

Course Code	Course Type	Course Name	Teaching Scheme			Examination Assessment Scheme				Credit scheme			
			Lecture	Tutorial	Practical	CA	End Sem	Practical	Total	L	T	P	C
BCA-CA-501	Major	Introduction to Data Science	3	0	4	40	60	100	200	3	0	4	5
BCA-CA-502	Major	Formal Languages and Automata Theory	4	0	0	40	60	-	100	4	0	0	4
BCA-CA-503	Major	Data Communication Networks	3	0	4	40	60	100	200	3	0	4	5
PCC-CA-501	VA	Financial Education & Investment Awarness	1	0	2	20	30	-	50	1	0	2	2
PEC-CA-501	DSE	A: DevOps B: Cloud (AWS/Azure) C: Blockchain D:NoSQL	2	0	4	40	60	100	200	2	0	4	4
HSMC-CA-501	AEC	Ability Enhancement	2	0	0	50	-	-	50	2	0	0	2
			15	**0**	**14**	**230**	**270**	**300**	**800**	**15**	**0**	**14**	**22**

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Third Year (5th SEM) BCA (2024-25 Course)

BCA-CA-501 : Introduction to Data Science

Teaching Scheme:	Credit	Examination Scheme:
TH: 4 Hours/Week	**3**	**Internal (TH): 40 Marks** **External (TH): 60 Marks**

Prerequisite Courses, if any:

- Fundamentals of Data Science
- Basic of mathematics, Matrices, Calculus and Statistics

Companion Course, if any: NO

Course Objectives:

- To focus on the analysis of data to extract knowledge and insight.
- Demonstrate proficiency with the methods and techniques for obtaining, organizing, exploring, and analyzing data.
- Learn skills to analyze real time problems using R.
- Recognize how data analysis, inferential statistics, modeling, machine learning, and statistical computing can be utilized in an integrated capacity.
- Create and modify customizable tools for data analysis and visualization per the evaluation of characteristics of the data and the nature of the analysis.
- Demonstrate the ability to clean and prepare data for analysis and assemble data from a variety of sources. To introduce the data science

Course Outcomes:

On completion of the course, learner will be able to–

CO21: Recognize various disciplines that contribute to a successful data science effort.

CO22: Be comfortable using commercial and open source tools such as the R language and its associated libraries for data analytics and visualization.

CO23: Able to use basic R - data structures in loading, cleaning the data and preprocessing the data.

CO24: Understand the processes of data science - identifying the problem to be solved, data collection, preparation, modeling, evaluation and visualization.

CO25: Able to cognizant of ethical issues in many data science tasks and its associated libraries for data analytics and visualization.

CO26: Able to do the exploratory data analysis on real time datasets to understand and implement lists, vectors, matrices, data frames, linear regression.

Course Contents

Unit I	Introduction	(04 Hours)
Introduction to Data Science – Evolution of Data Science – Data Science Roles – Stages in a Data Science Project – Applications of Data Science in various fields – Data Security Issues. Types of Data, Classification of digital Data with Example Applications. Sources of Data and Data Evolution		
Mapping of Course Outcomes for Unit I	CO1	
Unit II	**Introduction to R in Data Science**	**(06 Hours)**
Features of R - Environment - R Studio. Basics of R-Assignment - Modes - Operators - special numbers - Logical values - Basic Functions - R help functions - R Data Structures - Control Structures. Vectors: Definition- Declaration - Generating - Indexing - Naming - Adding & Removing elements - Operations on Vectors - Recycling - Special Operators - Vectorized if- then else-Vector Equality – Functions for vectors - Missing values - NULL values - Filtering & Sub setting.		
Mapping of Course Outcomes for Unit II	CO2, CO3	
Unit III	**Data Collection and Data Pre-Processing**	**(04 Hours)**
Data Collection Strategies – Data Pre-Processing Overview – Data Cleaning – Data Integration and Transformation – Data Reduction – Data Discretization.		
Mapping of Course Outcomes for Unit III	CO4	
Unit IV	**Exploratory Data Analytics**	**(04 Hours)**
Descriptive Statistics – Mean, Standard Deviation, Skewness and Kurtosis – Box Plots – Pivot Table – Heat Map – Correlation Statistics – ANOVA.		
Mapping of Course Outcomes for Unit IV	CO3, CO4	

Unit V	Model Development	(06 Hours)
Simple and Multiple Regression – Model Evaluation using Visualization – Residual Plot – Distribution Plot – Polynomial Regression and Pipelines – Measures for In-sample Evaluation – Prediction and Decision Making.		
Mapping of Course Outcomes for Unit V	CO4, CO5	
Unit VI	**Model Evaluation**	**(06 Hours)**
Generalization Error – Out-of-Sample Evaluation Metrics – Cross Validation – Over fitting – Under Fitting and Model Selection – Prediction by using Ridge Regression – Testing Multiple Parameters by using Grid Search		
Mapping of Course Outcomes for Unit VI	CO6	

Learning Resources

Text Books:

11. An introduction to Data Science by Jeffrey Stanton
12. The Elements of Data Analytic Style by Jeff Leek
13. Exploratory Data Analysis with R, by Roger Peng
14. OpenIntro Statistics, by Diez, Barr, and Centinkaya-Rundel

Reference Books:

19. Jojo Moolayil, “Smarter Decisions : The Intersection of IoT and Data Science”, PACKT, 2016.
20. Cathy O’Neil and Rachel Schutt , “Doing Data Science”, O'Reilly, 2015.
21. David Dietrich, Barry Heller, Beibei Yang, “Data Science and Big data Analytics”, EMC 2013
22. Raj, Pethuru, “Handbook of Research on Cloud Infrastructures for Big Data Analytics”, IGI Global.

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12

CO1	3	1	3	1	-	-	-	-	1	-	-	1
CO2	3	2	3	1	-	-	-	-	1	-	-	-
CO3	3	2	3	1	-	-	-	-	1	-	-	1
CO4	3	2	3	1	1	-	-	-	1	-	-	1
CO5	3	2	3	1	1	-	-	-	1	-	-	1
CO6	3	2	3	1	1	-	-	-	1	-	-	1

BCA-CA-501-Introduction to Data Science

Teaching Scheme **Practical: 04 Hours/Week**		**Examination Scheme and Marks** **Internal: 40 Marks** **External: 60 Marks**

Companion Course:

Course Objectives:

- Demonstrate knowledge in applying system software and tools available in modern operating system of data science.
- To learn skills and analyze real time problems using R
- Introduce students to the collection. Preparation, analysis, modeling and visualization of data, covering both conceptual and practical issues.

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Identify and describe the methods and techniques commonly used in data science.
CO2: Be cognizant of ethical issues in many data science tasks.
CO3: Be comfortable using commercial and open source tools such as the R language and its associated libraries for data analytics and visualization.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended :- Windows

Programming tools recommended: - R language

Virtual Laboratory:

- https://www.iiitmk.ac.in/DAVirtalLab/

Part I

Suggested List of Laboratory Experiments/Assignments (6 assignments are compulsory)	
Sr. No.	**Group A**
72.	Installing R and R studio. Then, create a folder DS_R and make it a working directory. Display the current working directory.
73.	Basic operations in R language and working with Vectors.
74.	Implementation of Reading data from the files and writing output back to the specified file.
75.	Implementation of Data Frame and its corresponding operators and functions.
76.	Implementation of Visualizations - Bar, Histogram, Box, Line, scatter plot, etc.
77.	Implementation of Linear and multiple Linear Regression.
	Group B (Mini Project) **Select any one problem statement**
6	Building Chatbots.
6	Credit Card Fraud Detection
6	Fake News Detection
6	Forest Fire Prediction
7	Classifying Breast Cancer
7	Driver Drowsiness Detection
7	Recommender Systems

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	2	2	3	-	-	-	-	-	-	1
CO2	1	2	2	2	2	-	-	-	-	-	-	1

CO3	1	2	2	2	2	-	-	-	-	-	-	1
CO4												
CO5												

Dr. D. Y. Patil Vidyapeeth, Pimpri, Pune

Dr. D. Y. Patil School of Science & Technology

Third Year BCA(2024-25 Course)

BCA-CA-303 : Data Communication and Networks

Teaching Scheme:	**Credit**	**Examination Scheme:**

TH: 03 Hours/Week **PR: 04 Hours/Week**	05	**Internal (TH): 40 Marks** **External (TH): 60 Marks**

Prerequisite Courses, if any:

- Fundamentals of Digital Electronics
- Basic of Computer Networking, data communication

Companion Course, if any: Computer Networks

Course Objectives:

- To Understand the Architectural Overview of Internet.
- To Understand the use of network components.
- To Understand the various Error detection and correction methods.
- Understand the architecture and its components and working of OSI and TCP/IP models and its performance.
- To Explore and learn about IP v4 and IPv6 addresses.

Course Outcomes:

On completion of the course, learner will be able to–

CO27: To understand process of data communication along with analog and digital signals, various network architectures.

CO28: To understand various types of guided and unguided media for transmission of data.

CO29: To use different types of error detection and correction methods in wireless communication.

CO30: To study and design different types of topologies using network components.

CO31: To analyze use of OSI reference model for networking.

CO32: To use appropriate IP addressing as per network condition, its architecture and use(IPv4 and IPv6).

Course Contents

Unit I	Fundamentals of Data Communication and Computer Networks	(06 Hours)

Process of data communication and its components: transmitter, receiver, medium, message, protocol. Protocol Standards, Standard of analog and digital signals organizations. Bandwidth, Data Transmission Rate, Baud Rate and Bits per second. Modes of Communication (Simplex, Hall duplex, Full Duplex). Analog Signal and Digital Signal, Analog and Digital transmission: Analog to Digital, Digital to Analog Conversion. Fundamental Of Computer Network: Definition and Need of Computer Network, Applications, Network Benefits. Classification of networks: LAN, MAN, WAN. Network architecture: Peer to Peer, Client-Server Network.

#Exemplar/Case Studies	Demonstration of LAN, peer to peer and client server network.
Mapping of Course Outcomes for Unit I	CO1

Unit II	Transmission Media and Switching	(06 Hours)

Communication Media: Guided Transmission Media, Twisted Pair Cable, Coaxial Cable, Fiber Optic Cable. Unguided Transmission Media, Radio Waves, Microwaves, Infrared, Satellite. Line-of-Sight Transmission Point to Point, Broadcast. Multiplexing: Frequency-Division multiplexing and Time-Division Multiplexing. Switching: Circuit-switched networks and Packet-switched networks.		
#Exemplar/Case Studies	Demonstration of various cables.	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**Error Detection, Correction and Wireless Communication**	**(06 Hours)**
Types of Errors: Single Bit Error and Burst Error, Redundancy. Error Detection: Longitudinal Redundancy Check (LRC), Vertical Redundancy Check (VRC), Cyclic Redundancy Check (CRC), Forward error Correction. IEEE standards: 802.1, 802.2, 802.3, 802.4, 802.5. Wireless LANs: 802.11 Architecture, MAC Sublayer, Addressing Mechanism. Bluetooth Architecture: Piconet, Scatternet. Mobile Generations: 1G, 2G, 3G, 4G and 5G.		
#Exemplar/Case Studies	Use of error detection and correction mechanism, Bluetooth mechanism.	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**Network Topologies and Network Devices**	**(06 Hours)**
Network Topologies: Introduction, Definition, Selection Criteria, Types of Topology- i) Bus ii) Ring iii) Star iv) Mesh v) Tree vi) Hybrid. Network Connecting Devices: Hub, Switch, Router, Repeater, Bridge, Gateway, Modern, Wireless infrastructure Components.		
#Exemplar/Case Studies	Demonstration of various topologies and network devices.	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Reference Models**	**(06 Hours)**
OSI Reference Model: Layered Architecture, Peer-to- Peer Process, Interfaces between Layer, Protocols, Organization of the Layers, Encapsulation Layers of the OSI Reference Model (Functions and features of each Layer) - Physical Layer, Data Link Layer, Network Layer, Transport Layer, Session Layer, Presentation Layer, Application Layer. TCP/IP model: Layered Architecture, Data link layer: nodes and links, services, categories of links, sublayers, Link layer addressing: three types of addresses, address resolution protocol (ARP). Network Layer Addresses: address space, classful and classless addressing, dynamic host configuration protocol (DHCP), network address resolution (NAT). Transport layer protocol: transport layer services, connectionless and connection-oriented protocol.		

#Exemplar/Case Studies	Uses of OSI model	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**Introduction to IPv4 and IPv6**	**(06 Hours)**
Introduction: Addressing mechanism in the Internet IP Addressing — IP Address classes, classless IP addressing, Subnetting, supernetting, Masking. IPv4 and IPv6 with all format details.		
#Exemplar/Case Studies	Demonstration of IPv4 and IPv6 addresses.	
Mapping of Course Outcomes for Unit VI	CO6	

Learning Resources

Text Books:

- Forouzan Behrouz A. - Data communications and networking|| Tata McGraw Hill, New Delhi , 2006, ISBN : 0070472971.
- Tanenbaum Andrew S. - Computer Networks || PHI Learning Pvt Ltd, Delhi , ISBN-13: 978-0-13-2126953.

Reference Books:

- Godbole Achyut - Data communications and networks|| Tata McGraw Hill, New Delhi, ' 2006, ISBN : 0070472971.
- Comer Douglas E - Internetworking with TCP/IP Principles, Protocols and Architectures|| PHI Learning Pvt Ltd, Delhi, ISBN: 81-203-2065-4.

Practical List:

1. Prepare specification table for guided and unguided media
2. Classify network connecting devices with their specifications.
3. Create a small Network. Install, configure various devices and perform at least one peer -to-peer service and client/server service over it.
4. Design layout of a Network for department, Deciding upon type of network, number/length of components with their specifications.
5. Interconnect two PCs using RS232 cable. Write step by step procedure to transfer a file from one computer to another through RS232 and implement.
6. Prepare hardware specification to develop a wireless LAN for a cyber café for 20 users.
7. Create a Bluetooth network of 5 devices namely laptop, mobile phone, speaker, printer, keyboard and transfer file from one device to other. Configure your laptop/mobile as a hotspot for internet access.
8. Prepare a proposal to develop a network system that links two branch offices of an organization. Two branches are separated by a distance of 10KM. Make appropriate assumption while making proposal.

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	3	1	-	-	-	-	1	-	-	1
CO2	3	2	3	1	-	-	-	-	1	-	-	-
CO3	3	2	3	1	-	-	-	-	1	-	-	-
CO4	3	2	3	1	1	-	-	-	1	-	-	-
CO5	3	2	3	1	1	-	-	-	1	-	-	1
CO6	3	2	3	1	1	-	-	-	1	-	-	1

Dr. D. Y. Patil Vidyapeeth, Pimpri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year of BCA (2024-25 Course)

BCA-CA-303 : Data Communication and Networks Lab

Teaching Scheme Practical: 02 Hours/Week	Credit Scheme 04	Examination Scheme and Marks Internal: 40 Marks External: 60 Marks

Companion Course: Computer Networks

Course Objectives:

- To Understand the Architectural Overview of Internet.
- To Understand the use of network components.
- To Understand the various Error detection and correction methods.
- Understand the architecture and its components and working of OSI and TCP/IP models and its performance.
- To Explore and learn about IP v4 and IPv6 addresses.

Course Outcomes:

On completion of the course, learner will be able to–

- CO1: To understand various types of guided and unguided media for transmission of data.
- CO2: To study and design different types of topologies using network components.
- CO3: Use of technical knowledge and skills to use IPV4 or IPV6 addresses to create a network.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based

on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Virtual Laboratory:

- http://cse18- iiith.vlabs.ac.in/Introduction.html?domain=Computer%20Scie nce
- http://vlabs.iitb.ac.in/vlabs-dev/labs/cglab/index.php

Part I : Data Communication and Networks Lab

Suggested List of Laboratory Experiments/Assignments

(6 assignments are compulsory)

Sr. No.	Group A
78.	Prepare specification table for guided and unguided media
79.	Classify network connecting devices with their specifications.
80.	Create a small Network. Install, configure various devices and perform at least one peer -to-peer service and client/server service over it.
81.	Design layout of a Network for department, Deciding upon type of network, number/length of components with their specifications.

82.	Interconnect two PCs using RS232 cable. Write step by step procedure to transfer a file from one computer to another through RS232 and implement.
83.	Prepare hardware specification to develop a wireless LAN for a cyber café for 20 users.
	Group B (Mini Project) **Select any one problem statement**
7	Create a Bluetooth network of 5 devices namely laptop, mobile phone, speaker, printer, keyboard and transfer file from one device to other. Configure your laptop/mobile as a hotspot for internet access.
7	Prepare a proposal to develop a network system that links two branch offices of an organization. Two branches are separated by a distance of 10KM. Make appropriate assumption while making proposal.

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	2	2	3	-	-	-	-	-	-	1
CO2	1	2	2	2	2	-	-	-	-	-	-	1
CO3	1	2	2	2	2	-	-	-	-	-	-	1

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year of Bachelor of Science (2024-25 Course)

PCC-CA-501: Financial Education and Investment Awareness

Teaching Scheme:	Credit	Examination Scheme:
TH: 01 Hours/Week **PR: 02 Hours/Week**	**02**	**Internal (TH): 20 Marks** **External (TH): 30 Marks**

Prerequisite Courses, if any:

- Students must have knowledge about mathematics and statistics.

Companion Course, if any: - PCC-BCS-301 Project Management

Course Objectives:

- To understand the operational nuances of a Finance Manager
- Comprehend the technique of making decisions related to finance function

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Develop various portfolio models
CO2: Develop fast, efficient and accurate excel skills
CO3: Recognize efficient financial budgeting and forecasting techniques
CO4: Familiarize the students with the valuation modelling of securities
CO5: Design and construct useful and robust financial modelling applications
CO6: Identify the business opportunities.

Course Contents

Unit I	FOUNDATIONS OF FINANCE	(03 Hours)
Financial management – An overview- Time value of money- Introduction to the concept of risk and return of a single asset and of a portfolio- Valuation of bonds and shares-Option valuation.		
#Exemplar/Case Studies	Fundamental Analysis of Bharat Electronics Limited	
Mapping of Course Outcomes for Unit I	CO1	

Unit II	INVESTMENT DECISIONS	(05 Hours)
Capital Budgeting: Principles and techniques - Nature of capital budgeting- Identifying relevant cash flows - Evaluation Techniques: Payback, Accounting rate of return, Net Present Value, Internal Rate of Return, Profitability Index - Comparison of DCF techniques - Project selection under capital rationing - Inflation and capital budgeting - Concept and measurement of cost of capital - Specific cost and overall cost of capital.		
#Exemplar/Case Studies	Adidas: Sustainability Bond	

Mapping of Course Outcomes for Unit II	CO2	
Unit III	**FINANCING AND DIVIDEND DECISION**	**(04 Hours)**
Financial and operating leverage - capital structure - Cost of capital and valuation – designing capital structure. Dividend policy - Aspects of dividend policy - practical consideration - forms of dividend policy - forms of dividends - share splits		
#Exemplar/Case Studies	Johnson & Johnson Spin-off	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**WORKING CAPITAL MANAGEMENT**	**(04 Hours)**
Principles of working capital: Concepts, Needs, Determinants, issues and estimation of working capital - Accounts Receivables Management and factoring - Inventory management – Cash management - Working capital finance : Trade credit, Bank finance and Commercial paper.		
#Exemplar/Case Studies	Reliance`s Foreign Currency Bond	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**LONG TERM SOURCES OF FINANCE**	**(03 Hours)**
Indian capital and stock market, New issues market Long term finance: Shares, debentures and term loans, lease, hire purchase, venture capital financing, Private Equity.		
#Exemplar/Case Studies	Tesla`s Convertible Bonds	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**BUSINESS AND FINANCING**	**(05 Hours)**
Small Enterprises – Characteristics, Ownership Structures – Steps involved in setting up a Business – identifying, selecting a Good Business opportunity, Market Survey and Research, Techno Economic Feasibility Assessment – Preparation of Preliminary Project Reports – Project 100 Appraisal –Sources of Finance, Management of working Capital, Costing, Break Even Analysis, Taxation – Income Tax, Excise Duty – Sales Tax.		
#Exemplar/Case Studies	Stock Split: A Case of IRCTC	
Mapping of Course Outcomes for Unit VI	CO6	

Learning Resources

TEXT BOOKS:

1. M.Y. Khan and P.K. Jain Financial management, Text, Problems and cases Tata McGraw Hill, 6th edition, 2011.
2. M. Pandey Financial Management, Vikas Publishing House Pvt. Ltd., 10th edition, 2012.

Reference Books:

1. Aswat Damodaran, Corporate Finance Theory and practice, John Wiley & Sons, 2011.
2. James C. Vanhorne –Fundamentals of Financial Management– PHI Learning, 11th Edition, 2012.
3. Brigham, Ehrhardt, Financial Management Theory and Practice, 12th edition, Cengage Learning 2010.
4. Prasanna Chandra, Financial Management, 9th edition, Tata McGraw Hill, 2012.
5. Srivatsava, Mishra, Financial Management, Oxford University Press, 2011.

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	3	-	-	-	-	-	-	-	-	-
CO2	1	1	-	-	2	-	-	-	1	-	-	1
CO3	2	-	1	-	1	-	-	-	2	-	2	-
CO4	-	-	2	-	-	-	-	-	-	-	2	-
CO5	2	1	-	-	2	-	-	-	-	-	-	1
CO6	2	1	-	-	-	-	-	-	-	-	-	-

PCC-CA-501: Financial Education and Investment Awareness

Teaching Scheme	Credit Scheme	Examination Scheme and Marks
PR: 02 Hours/Week	02	Internal (PR): 40 Marks External (PR): 60 Marks

Companion Course: PEC-CA-401: Software Project Management

Course Objectives:

- To understand the operational nuances of a Finance Manager
- Comprehend the technique of making decisions related to finance function

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Recognize efficient financial budgeting and forecasting techniques
CO2: Familiarize the students with the valuation modelling of securities
CO3: Design and construct useful and robust financial modelling applications
CO4: Identify the business opportunities.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp

of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended: - Windows / Linux

Programming tools recommended: MS Excel

Part I : Financial Education Lab

Suggested List of Laboratory Experiments/Assignments

(6 assignments are compulsory)

Sr. No.	Group A
84.	Case study on Implementing a Zero Debtor Policy through Channel Financing in an MNC
85.	An Analysis of Small Savings Schemes in India
86.	Analyzing the Risk Weighted Performance of Equity Mutual Funds
87.	Study of Efficient Market Hypothesis: Evidence from Bonus Issue
88.	Study of The Microfinance Industry in India
89.	Study of Maruti Udyog's Accounting Policies
90.	Study of Film Insurance & Financing in India
91.	Study of Coimbatore Bypass Road Project
	Group B (Mini Project) **Select any one problem statement**
7	**ICICI Prudential Life Insurance – The Importance of a Strong Brand Image**

7	**A Study on Mergers and Acquisitions in the Indian Banking Sector**
7	**Evaluating the Capital at South Central Railway**
7	**Evaluating the Performance of Private and Public Mutual Funds**
7	**The Impact of Demonetization on Tourism in Goa**
8	**Comparative Analysis of Regional Rural Banks in India**

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2		2	3	-	-	-	-	-	-	1
CO2	1		2	2	2	-	-	-	-	-	-	1
CO3	1	2	2	2	2	-	-	-	-	-	-	1
CO4	2	1		1	1							

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Third Year of Bachelor of Science (Computer Science) (2024-25 Course)

PEC-CA-501 : Discipline Specific Elective-6

Teaching Scheme:	Credit	Examination Scheme:
TH: 2 Hours/Week	**4**	**Internal (TH): 20 Marks** **External (TH): 30 Marks**

Prerequisite Courses, if any:

- For planning: Jira
- For building: Maven, Gradle, Docker, Github, Gitlab
- For Continuous integration: Jenkins, Travis CI

Companion Course, if any: Embedded Systems and IoT

Course Objectives:

- Understand the key concepts and principles of DevOps
- List the most common DevOps tools
- Identify the business benefits of DevOps and continuous delivery.
- Recall the specific DevOps methodologies and frameworks

Course Outcomes:

On completion of the course, learner will be able to–
CO1: Describe the evolution of technology & timeline (Understand)
CO2: Explain Introduction to various Devops platforms (Remember)
CO3: Demonstrate the building components / blocks of Devops and gain an insight of the Devops Architecture. (Understand)
CO4: Apply the knowledge gain about Devops approach across various domains (Apply)
CO5: Build DevOps application (Apply)

Course Contents

Unit I	**Introduction to DevOps.**	**(04 Hours)**

Introduction to DevOps.-Define Devops ,What is Devops,SDLC models, Lean, ITIL, Agile,Why Devops? , History of Devops,Devops Stakeholders,Devops Goals,Important terminology,Devops perspective,DevOps and Agile,DevOps Tools,Configuration management,Continuous Integration and Deployment,Linux OS Introduction,Importance of Linux in DevOps,Linux Basic Command Utilities,Linux Administration,Environment Variables,Networking,Linux Server Installation,RPM and YUM Installation

#Exemplar/Case Studies	**Continuous Compliance Monitoring**
Mapping of Course Outcomes for Unit I	CO1

Unit II	**Version Control-GIT**	**(03 Hours)**

Introduction to GIT,What is Git,About Version Control System and Types ,Difference between CVCS and DVCS ,A short history of GIT,GIT Basics ,GIT Command Line,Installing Git ,Installing on Linux , Installing on Windows , Initial setup,Git Essentials,Creating repository,Cloning, check-in and committing,Fetch pull and remote , Branching,Creating the Branches, switching the branches, merging,The branches.

#Exemplar/Case Studies	**GitHub.com**
Mapping of Course Outcomes for Unit II	CO2, CO3

Unit III	**Chef for configuration management**	**(13 Hours)**

Chef for configuration management- Overview of Chef; Common Chef Terminology (Server, Workstation, Client, Repository Etc.) Servers and Nodes Chef Configuration Concepts. Workstation Setup: How to configure knife Execute some commands to test connection between knife and workstation., Organization Setup: Create organization; Add yourself and node to organization., Test Node Setup: Create a server and add to organization, check node details using knife., Node Objects and Search: How to Add Run list to Node Check node Details., Environments: How to create Environments, Add servers to environments. Roles: Create roles, Add Roles to organization., Attributes: Understanding of Attributes, Creating Custom Attributes, Defining in Cookbooks., Data bags: Understanding the data bags, Creating and managing the Data bags, Creating the data bags using CLI and Chef Console, Sample Data bags for Creating Users.

#Exemplar/Case Studies	MultiTier Development Application
Mapping of Course Outcomes for Unit III	CO2,CO3

Unit IV	**Build tool- Maven**	**(10 Hours)**

Build tool- Maven - Maven Installation,Maven Build requirements,Maven POM Builds (pom.xml),Maven Build Life Cycle,Maven Local Repository (.m2),Maven Global Repository ,Group ID, Artifact ID, Snapshot,Maven Dependencies,Maven Plugins

#Exemplar/Case Studies	TeamCity
Mapping of Course Outcomes for Unit IV	CO3, CO4

Unit V	**Docker– Containers & Build tool- Maven**	**(12 Hours)**

Docker– Containers & Build tool- Maven - Introduction: What is a Docker, Use case of Docker, Platforms for Docker, Dockers vs. Virtualization,Architecture: Docker Architecture., Understanding the Docker components, Installation: Installing Docker on Linux. Understanding Installation of Docker on windows. Some Docker commands. Provisioning. Docker Hub.: Downloading Docker images. Uploading the images in Docker Registry and AWS ECS, Understanding the containers, Running commands in container. Running multiple containers.,Custom images: Creating a custom image. Running a container from the custom image. Publishing the custom image, . Docker Networking: Accessing containers, linking containers, Exposing container ports, Container Routing.

#Exemplar/Case Studies	Healthcare
Mapping of Course Outcomes for Unit V	CO3, CO5,CO6

Learning Resources

Text Books:

15. DevOps For Beginners: A Complete Guide To DevOps Best Practices (Including How You Can Create World-Class Agility, Reliability, And Security In ... With DevOps): 2 (Code Tutorials)
    By Craig Berg, ISBN: 979-8653362941
16. Effective DevOps: - Building a Culture of Collaboration, Affinity, and Tooling at Scale (English, Paperback, Davis Jennifer), ISBN: 9789352133765, 9789352133765
17. DevOps For Dummies by Freeman , ISBN: 9788126553495

Reference Books:

23. DevOps for Developers: Michael Hüttermann
24. DevOps: A Software Architect's Perspective: Ingo M. Weber, Len Bass, and Liming Zhu
25. Building a DevOps Culture: Jennifer Davis, Katherine Daniels. Publisher: O'Reilly
26. Practical DevOps: Joakim Veronal
27. DevOps for Dummies: Gene Kim, Kevin Behr, George, Publisher: John Wiley & Sons

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	1	1	-	-	-	-	1	-	-	1
CO2	3	3	3	3	1	1	1	1	1	3	3	1
CO3	3	2	3	1	3	1	1	1	1	3	3	1
CO4	3	2	3	1	1	1	3	1	1	1	1	1
CO5	3	3	3	1	1	-	-	-	1	-	-	1

Discipline Specific Elective -6 DevOPsLab

Teaching Scheme	Credit Scheme	Examination Scheme and Marks
Practical: 02 Hours/Week	**04**	**Internal: 40 Marks** **External: 60 Marks**

Companion Course: PEC-CA-501: Discipline Specific Elective -6 (DevOps)

Course Objectives:

- To understand DevOps practices which aims to simplify Software Development Life Cycle
- To be aware of different Version Control tools like GIT, CVS or Mercurial
- To Integrate and deploy tools like Jenkins and Maven, which is used to build, test and deploy applications in DevOps environment
- To be familiarized with selenium tool, which is used for continuous testing of applications deployed.
- To use Docker to Build, ship and manage applications using containerization
- To understand the concept of Infrastructure as a code and install and configure Ansible tool

Course Outcomes:

On completion of the course, learner will be able to–

CO1: To understand the fundamentals of DevOps engineering and be fully proficient with DevOps terminologies, concepts, benefits, and deployment options to meet your business requirements CO2: To obtain complete knowledge of the "version control system" to effectively track changes augmented with Git and GitHub.
CO3: To understand the importance of Jenkins to Build and deploy Software Applications on server environment.
CO4: Understand the importance of Selenium and Jenkins to test Software Applications.
CO5 : To understand concept of containerization and Analyze the Containerization of OS images and deployment of applications over Docker and To understand concept of containerization and Analyze the Containerization of OS images and deployment of applications over Docker

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended: - Windows

Programming tools recommended: - Android

Virtual Laboratory:

Part I: DevOps Lab

Suggested List of Laboratory Experiments/Assignments

(8 assignments are compulsory)

Sr. No.	Group A
1.	To understand DevOps: Principles, Practices, and DevOps Engineer Role and Responsibilities.
2.	To understand Version Control System / Source Code Management, install git and create a GitHub account.
3.	To Perform various GIT operations on local and Remote repositories using GIT Cheat-Sheet
4.	To Setup and Run Selenium Tests in Jenkins Using Maven.
5.	To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job.
6.	To understand Docker Architecture and Container Life Cycle, install Docker and execute docker commands to manage images and interact with containers.
7.	To Build the pipeline of jobs using Maven / Gradle / Ant in Jenkins, create a pipeline script to Test and deploy an application over the tomcat server.
8.	To understand Jenkins Master-Slave Architecture and scale your Jenkins standalone implementation by implementing slave nodes.
	Group B
8	To install and Configure Pull based Software Configuration Management and provisioning tools using Puppet.
8	To provision a LAMP/MEAN Stack using Puppet Manifest.

8	To learn Software Configuration Management and provisioning using Puppet Blocks(Manifest, Modules, Classes, Function) (Mini Project)

@The CO-PO Mapping Matrix

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	1	1	-	-	-	-	1	-	-	1
CO2	3	3	3	3	1	1	1	1	1	3	3	1
CO3	3	2	3	1	3	1	1	1	1	3	3	1
CO4	3	2	3	1	1	1	3	1	1	1	1	1
CO5	3	3	3	1	1	-	-	-	1	-	-	1

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year of BACHELOR OF SCIENCE (COMPUTER SCIENCE)

(2024-25 Course)

PEC-CA-501 A: Discipline Specific Elective -3 (Cloud(AWS/Azure))

Teaching Scheme:	Credit	Examination Scheme:
TH: 2 Hours/Week	4	**Internal (TH): 20 Marks** **External (TH): 30 Marks**

Prerequisite Courses, if any:

- Fundamentals of Embedded Systems, IoT
- Basic of Computer Networking, data communication ,

Companion Course, if any: Embedded Systems and IoT

Course Objectives:

6. Describe the AWS Cloud and the AWS global infrastructure
7. Recognize and explain basic AWS Cloud architectural principles
8. Describe key services on the AWS platform and their common use cases
9. Describe the basic security and compliance aspects of the AWS platform and the shared security model
10. Define the billing, account management, and pricing models
11. Describe basic/core characteristics of deploying and operating in the AWS Cloud
12. To understand the azure virtual machines
13. Recognize the services offered by Azure
14. Understand the azure storage
15. Configure the Azure active directory services To understand the azure virtual machines
16. Recognize the services offered by Azure
17. Understand the azure storage
18. Configure the Azure active directory services

Course Outcomes:

On completion of the course, learner will be able to–

CO1:Windows Azure Account and IAAS, PAAS, SAAS on Aws Cloud platform (Creation & Apply)

CO2 :Virtual Machine on Server Application (Plan)

CO3: Virtual Machine to cluster and deployment of load balances and Managing Voluminous Information with EBS, Glacier Storage Service (Understand)

CO4: Interpret Architecture and Pharrell Programing of Cloud Computing. (Apply)

CO5: Demonstrate practical implementation of Cloud computing and to understand the azure virtual machines . and Amazon Identity and Access Management ,Internet Gateway in Cloud Platform (Understand)

Course Contents

Unit I	Introduction to Microsoft Azure Virtual machines	(09 Hours)

Introduction to Microsoft Azure Virtual machines: Introduction to Azure VM - Resource planning with Basic and standard vm - VM pricing - Difference between basic and standard vm - Creating virtual machines - Choosing the type of vm - Configuring DNS address - Configuring endpoints - Connecting to virtual machine - Implementing the lifecycle of a virtual machine - Uploading and downloading virtual hard disks - Attaching an empty hard disk to vm - Creating VM from a custom image - Deleting images and disks

#Exemplar/Case Studies	Food Service

Mapping of Course Outcomes for Unit I	CO1,CO2	
Unit II	**Azure Networking**	**(09 Hours)**
Azure Networking : Creating and configuring a virtual network - Deploying a virtual machine in a virtual network - Deploying a web service in a virtual network - Modifying a network configuration - Configuring access control list - Configuring reserved IP addresses - Configuring public IP addresses - Implementing a point-to-site VPN - Implementing a site-to-site VPN - Implementing a virtual network to virtual network vpn - Configuring internal load balancing		
#Exemplar/Case Studies	Payment Gateway	
Mapping of Course Outcomes for Unit II	CO2, CO3	
Unit III	**Azure Storage**	**(09 Hours)**
Azure Storage : Storage account in azure - Implement blobs and azure files - Types of storage in azure - Blob - Table - Queue - Drives - Managing storage account keys - Implementing SQL databases - Choosing a service tier - Implementing point-in-time recovery - Implementing georeplication - Scalability strategies - Importing and exporting data.		
#Exemplar/Case Studies	Media BroadCasting Service	
Mapping of Course Outcomes for Unit III	CO2,CO3	
Unit IV	**Introduction to Cloud Computing And Amazon Web Services**	**(09 Hours)**
Introduction to Cloud Computing And Amazon Web Services: Introduction to Cloud Computing, Cloud Service Delivery Models (IAAS, PAAS, SAAS), Cloud Deployment Models (Private, Public, Hybrid and Community), Cloud Computing Security, Case Study Introduction to Amazon Web Services, Why Amazon? Use Cases, AWS Storage Options, AWS Compute Options, AWS Database Options, AWS Workflow Automation and Orchestration Options, AWS Systems Management And Monitoring Options, AWS Virtual Private Cloud Introduction, Pricing Concepts.		
#Exemplar/Case Studies	Pricing Model: Usage Reporting, billing andmetering (AWS), Cloud Statistics	
Mapping of Course Outcomes for Unit IV	CO3, CO5	

Unit V	AWS Storage	(09 Hours)
AWS Storage: Amazon Storage, S3 Storage Basics, Buckets and Objects, Creating A Web Server Using S3 Endpoints, Managing Voluminous Information with EBS, Glacier Storage Service , Describe Amazon Dynamo, Understand key aspects of Amazon RDS, Launch an Amazon RDS instance		
#Exemplar/Case Studies	Cryon	
Mapping of Course Outcomes for Unit V	CO3, CO5	
Unit VI	**AWS Networking**	**(09 Hours)**
AWS Networking: Introduction to AWS Networking , Access Control Lists (ACLs), Setting Up a Security Group, Setting Up VPC And Internet Gateway, Setting Up A VPN, Setting Up A Customer Gateway For VPN, Setting Up Dedicated Hardware For VPC, Scenario 1:VPC With A Public Subnet Only (Standalone Web), Scenario 2: VPC with Public And Private Subnets (3 Tier App), Scenario 3:VPC With Public And Private Subnets And Hardware VPN Access (Web On The Cloud, Database and App On Prem) Scenario 4: VPC With A Private Subnet Only And Hardware VPN Access. (Extension Of Your Corporate Network), Route53 for 9 SUB DNS System, Cloud front, Case Study		
#Exemplar/Case Studies	Xebia	
Mapping of Course Outcomes for Unit VI	CO4,CO5	

Learning Resources

Text Books:

18. Microsoft Azure Essentials: Fundamentals of Azure, 2nd Edition, Michael Collier, Robin Shahan ISBN: 978-1-5093-0296-3.

1. AWS Certified Solutions Architect Official Study Guide: Associate Exam (Aws Certified Solutions Architect Official: Associate Exam) by Joe Baron,Hisham Baz , Tim Bixler , Biff Gaut, Kevin E. Kelly ISBN: 978-1119138556

Reference Books:

1. Cloud Computing Bible by Barrie Sosinsky, Wiley India Pvt. Ltd,
2. Microsoft Azure Essentials: Fundamentals of Azure (ISBN 9780735697225), Michael S. Collier and Robin E. Shahan
3. Microsoft Azure Essentials: Fundamentals of Azure (ISBN 9780735697225),
4. Yohan Wadia , "AWS Certified Solutions Architect Official Study Guide: Associate Exam, John Packt Publishing
5. Bernald Golden, "Amazon Web Services for Dummies", John Wiley & Sons.

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	2	1	2	3	-	-	1	-	-	1
CO2	3	1	3	1	3	3	-	-	1	-	-	-
CO3	3	2	3	2	3	2	2	-	1	-	-	-
CO4	3	2	3	2	1	2	-	-	1	-	-	-
CO5	3	2	3	1	1	1	1	-	1	-	-	1

PEC-CA-301A: Cloud AWS, AZURE Lab

Teaching Scheme		Examination Scheme and Marks
Practical: 02 Hours/Week		**Internal: 40 Marks** **External: 60 Marks**

Companion Course: ESC-CS 601: Cloud Computing

Course Objectives:

- Describe the AWS Cloud and the AWS global infrastructure
- Recognize and explain basic AWS Cloud architectural principles
- Describe key services on the AWS platform and their common use cases
- Describe the basic security and compliance aspects of the AWS platform and the shared security model
- Define the billing, account management, and pricing models
- Describe basic/core characteristics of deploying and operating in the AWS Cloud
- To understand the azure virtual machines
- Recognize the services offered by Azure
- Understand the azure storage
- Configure the Azure active directory services

Course Outcomes:

On completion of the course, learner will be able to–

CO1 : : IAAS, PAAS, SAAS on Aws Cloud platform and Monitoring Azure Services (Apply)
CO2 : EC2 instances from of AMI's and Windows Azure Account (Creation)
CO3 : Managing Voluminous Information with EBS, Glacier Storage Service and Virtual Machine on ServerApplication (Plan)
CO4 : Amazon Identity and Access Management(Understand)
CO5 : VPC And Internet Gatewayin Cloud Plarform and Monitoring Azure Services (Plan)

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended :- Ghost OS,CloudMe

Programming tools recommended: - CloudZero,Amazon Web Services,Google App Engine

Virtual Laboratory:

- https://vlab.noaa.gov/web/osti-modeling/cloud-computing1
- https://www.codio.com/solutions/virtual-labs

Part I : Cloud Computing	
Suggested List of Laboratory Experiments/Assignments **(6 assignments are compulsory)**	
Sr. No.	**Group A**
1.	AWS root user account creation using AWS management console
2.	Understanding AWS Billing Dashboard and Setting up billing alerts using Cloud Watch
3.	Launching an EC2 instance and accessing it through SSH using putty.
4.	Creating web server on EC2, with and without bash script
5.	Create and document the process of creating a windows azure account
6.	Create a virtual machine from available releases of windows server images
7.	Create a virtual machine using the option “quick Create”
8.	Create a custom VM and Capture the image
	Group B (Mini Project) **Select any one problem statement**
9.	Creating and hosting static web site using S3 bucket.
10	**Demonstrating Amazon SNS service.**
11	**Configuration of Database engine using Amazon RDS.**
12	**Creating DNS using Route 53**
13	**Create a SQL server DB , Create tables and add data to the table**
14	**Test basic sql commands on the table created in the previous step.**
15	**Migrate an on premise DB to Azure**
16	Create a storage account in Azure

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	1	1	1	1	-	1	1	-	-	1

CO2	1	3	1	3	1	-	1	-	1	-	-	-
CO3	2	2	3	1	2	1	2	1	1	-	-	-
CO4	2	3	1	1	1	2	1	-	1	-	-	-
CO5	1	1	1	2	1	2	2	1	1	-	-	1

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune **Dr. D. Y. Patil School of Science & Technology** **Fourth Year of Engineering (2024-25 Course)** **PEC-CA-501: Discipline Specific Elective -5 (Blockchain Technologies)**		
Teaching Scheme:	**Credit**	**Examination Scheme:**

TH: 2 Hours/Week	**4**	**Internal (TH): 20 Marks** **External (TH): 30 Marks**

Prerequisite Courses, if any:

- Knowledge of Data structures.
- Students must have knowledge of some programming languages (such as C, C++, and Java).

Companion Course, if any: Embedded Systems and IoT

Course Objectives:

- Understand how blockchain systems (mainly Bitcoin and Ethereum) work.
- Design, build, and deploy smart contracts and distributed applications.
- Integrate ideas from blockchain technology into their own projects.

Course Outcomes:

On completion of the course, learner will be able to–
CO1: Discuss the cryptographic building blocks of block chain Technology (Understand)
CO2: Explain the fundamental concepts of block chain Technology (Understand)
CO3: Summarize the classification of consensus algorithms(Understand)
CO4 :Explain the concepts of first decentralized crypto-currency Bitcoin(Understand)
CO5 :Explain the use of smart contracts and its use cases.(Understand)
CO 6 Develop simple applications using Solidity language on Ethereum platform(Apply)

Course Contents

Unit I	**Fundamentals of Cryptography**	**(08 Hours)**

Fundamentals of Cryptography -Introduction to Cryptography, Symmetric cryptography – AES. Asymmetric cryptography – RSA. Cryptographic hash functions-Applications of cryptographic hash functions – Merkle trees, Distributed hash tables.

#Exemplar/Case Studies	
Mapping of Course Outcomes for Unit I	CO1

Unit II	**Fundamentals of Block chain Technology**	**(08 Hours)**

Fundamentals of Block chain Technology - Block chain – Definition, architecture, elements of block chain, benefits and limitations, types of block chain. Consensus – definition, types, consensus in block chain. Decentralization – Decentralization using block chain, Methods of decentralization, Routes to decentralization, Block chain and full ecosystem decentralization.

#Exemplar/Case Studies	

Mapping of Course Outcomes for Unit II	CO2, CO3	
Unit III	**Consensus Algorithms and Bitcoin**	**(08 Hours)**
Consensus Algorithms and Bitcoin - Consensus Algorithms, Crash fault-tolerance (CFT) algorithms – Paxos, Raft. Byzantine fault-tolerance (BFT) algorithms – Practical Byzantine Fault Tolerance (PBFT), Proof of work (PoW), Proof of stake (PoS), Types of PoS. Bitcoin – Definition, Cryptographic keys – Private keys, public keys, addresses. Transactions – Lifecycle, coinbase transactions, transaction validation. Block chain – The genesis block. Mining – Tasks of miners, mining algorithm, hash rate. Wallets – Types of wallets.		
#Exemplar/Case Studies		
Mapping of Course Outcomes for Unit III	CO2,CO3	
Unit IV	**Smart Contracts and Use cases**	**(10 Hours)**
Smart Contracts and Use cases - Smart Contracts – Definition, Smart contract templates, Oracles, Types of oracles, Deploying smart contracts. Decentralization terminology – Decentralized applications, Decentralized Autonomous Organizations. Use cases of Block chain technology – Government, Health care, Finance, Supply chain management. Block chain and allied technologies – Block chain and Cloud Computing, Block chain and Artificial Intelligence.		
#Exemplar/Case Studies		
Mapping of Course Outcomes for Unit IV	CO3, CO4	
Unit V	**Ethereum and Solidity**	**(11 Hours)**
Ethereum and Solidity - Ethereum – The Ethereum network. Components of the Ethereum ecosystem – Keys and addresses, Accounts, Transactions and messages. The Ethereum Virtual Machine, Blocks and block chain. The Solidity language – The layout of a Solidity source code, Structure of a smart contract, variables, data types, control structures, events, inheritance, libraries, functions, error handling. Smart contracts .		
#Exemplar/Case Studies	Voting, Auction	
Mapping of Course Outcomes for Unit V	CO3, CO5,CO6	
Learning Resources		

Text Books:

- Mastering Blockchain, by Imran Bashir, II edition Packt Publications
- BlockChain: Blueprint for a new economy, by Melanie Swan O'Reilly Publications.
- Mark Gates, "Block chain: Ultimate guide to understanding block chain, bit coin, crypto currencies, smart contracts and the future of money", Wise Fox Publishing and Mark Gates 2017.
- Salman Baset, Luc Desrosiers, Nitin Gaur, Petr Novotny, Anthony O'Dowd, Venkatraman Ramakrishna, "Hands-On Block chain with Hyper ledger: Building decentralized applications with Hyperledger Fabric and Composer", 2018.
- Bahga, Vijay Madisetti, "Block chain Applications: A Hands-On Approach", Arshdeep Bahga, Vijay Madisetti publishers 2017.

Reference Books:

1. BlockChain: A Beginners Guide", Authors: SherminVoshmgir, Valentin Kalinov Publisher: https://blockchainhub.net/
2. "Cryptocurrency and Bitcoin Technologies", Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder published by Princeton University

19. Andreas Antonopoulos, "Mastering Bitcoin: Unlocking Digital Crypto currencies", O'Reilly Media, Inc. 2014. 2. Melanie Swa, "Block chain ",O'Reilly Media 2014.

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	-	-	-	-	-	-	-	-	-	-
CO2	2	2	2	-	-	-	-	-	-	-	-	1
CO3	2	2	3	-	-	-	-	-	-	-	-	1
CO4	3	2	3	-	-	-	-	-	-	-	-	1
CO5	3	3	3	1	1	-	-	-	-	-	-	1
CO 6	3	2	2	2	2	-	-	-	-	-	-	2

Discipline Specific Elective -5 Block Chain Lab

Teaching Scheme	Credit Scheme	Examination Scheme and Marks
Practical: 02 Hours/Week	**04**	**Internal: 40 Marks** **External: 60 Marks**

Companion Course: PEC-CA-501: Discipline Specific Elective -5 (Blockchain Technologies)

Course Objectives:

- Understanding Block chain Fundamentals and creating basic blocks.
- Able to Develop Block chain Applications in a structured manner
- Ability to create own crypto currency and get familiarity with future currencies.
- Able to Evaluate and Analyze Block chain Systems

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Knowledge of Blockchain Concepts and creating basic blocks.(Understand)
CO2: Proficiency in Blockchain Development.(Apply)
CO3: Ability to Design and Implement Blockchain Applications.(Implement)
CO4: Evaluation and Analysis of Blockchain Systems. (Apply)
CO5: Knowledge of crypto currency and creating a basic form of it.(Understand)

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based

on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended: - Windows

Programming tools recommended: - Android

Virtual Laboratory:

https://pranet.iith.ac.in/

Part I: Block chain Lab

Suggested List of Laboratory Experiments/Assignments

(8 assignments are compulsory)

Sr. No.	Group A
92.	Creating Merkle tree
93.	Creation of Block
94.	Blockchain implementation
95.	Creating ERC20 token
96.	Blockchain implementation using Merkle Trees
97.	Mining in Blockchain

98.	Peer-to-Peer implementation using Blockchain
99.	Creating Crypto-currency
	Group B
8	Develop block chain solutions and write smart contract Hyperledger Fabric and Ethereum frameworks
8	Crypto-currency Wallet (Mini Project)

@The CO-PO Mapping Matrix

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	1	1	-	-	-	-	1	-	-	1
CO2	3	3	3	3	1	1	1	1	1	3	3	1
CO3	3	2	3	1	3	1	1	1	1	3	3	1
CO4	3	2	3	1	1	1	3	1	1	1	1	1
CO5	3	3	3	1	1	-	-	-	1	-	-	1

Dr. D. Y. Patil Vidyapeeth, Pimpri, Pune

Dr. D. Y. Patil School of Science & Technology

Third Year (5th SEM) of BCA (2024-25 Course)

PEC-CA-501 : NoSQL

Teaching Scheme:	Credit	Examination Scheme:
TH: 2 Hours/Week	**4**	**Internal (TH): 40 Marks** **External (TH): 60 Marks**

Prerequisite Courses, if any:

- Basic Knowledge about DBMS

Companion Course, if any:

Course Objectives:

- Explore the origins of NoSQL databases and the characteristics that distinguish them from traditional relational database management systems.
- Understand the architectures and common features of the main types of NoSQL databases (key-value stores, document databases, column-family stores, graph databases).
- Discuss the criteria that decision makers should consider when choosing between relational and non-relational databases.
- Introducing the techniques for selecting the NoSQL database that best addresses specific use cases.

Course Outcomes:

On completion of the course, learner will be able to–

CO33: Explain the detailed architecture, Database properties and storage requirements
CO34: Differentiate and identify right database models for real time applications
CO35: Outline Key value architecture and characteristics
CO36: Design Schema and implement CRUD operations, distributed data operations
CO37: Compare data ware housing schemas and implement various column store internals
CO38: Choose and implement Advanced columnar data model functions for the real time applications

Course Contents

Unit I	INTRODUCTION TO NOSQL CONCEPTS	(06 Hours)

Data base revolutions: First generation, second generation, third generation, Managing Transactions and Data Integrity, ACID and BASE for reliable database transactions, Speeding performance by strategic use of RAM, SSD, and disk, Achieving horizontal scalability with database sharding, Brewers CAP theorem.

Mapping of Course Outcomes for Unit I	CO1

Unit II	NOSQL DATA ARCHITECTURE PATTERNS	(06 Hours)
NoSQL Data model: Aggregate Models- Document Data Model- Key-Value Data Model Columnar Data Model, Graph Based Data Model Graph Data Model, NoSQL system ways to handle big data problems, Moving Queries to data, not data to the query, hash rings to distribute the data on clusters, replication to scale reads, Database distributed queries to data nodes.		
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**KEY VALUE DATA STORES**	**(06 Hours)**
From array to key value databases, Essential features of key value Databases, Properties of keys, Characteristics of Values, Key-Value Database Data Modeling Terms, Key-Value Architecture and implementation Terms, Designing Structured Values, Limitations of Key Value Databases, Design Patterns for Key-Value Databases, Case Study: Key-Value Databases for Mobile Application Configuration.		
Mapping of Course Outcomes for Unit III	CO2, CO3	
Unit IV	**DOCUMENT ORIENTED DATABASE**	**(04 Hours)**
Document, Collection, Naming, CRUD operation, querying, indexing, Replication, Sharding, Consistency Implementation: Distributed consistency, Eventual Consistency, Capped Collection, Case studies: document oriented database: MongoDB and/or Cassandra.		
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**COLUMNAR DATA MODEL**	**(04 Hours)**
Data warehousing schemas: Comparison of columnar and row-oriented storage, Column-store Architectures: C-Store and Vector-Wise, Column-store internals and, Inserts/updates/deletes, Indexing, Adaptive Indexing and Database Cracking.		
Mapping of Course Outcomes for Unit V	CO5, CO6	
Unit VI	**DATA MODELING**	**(04 Hours)**
Advanced techniques: Vectorized Processing, Compression, Write penalty, Operating Directly on Compressed Data Late Materialization Joins, Group-by, Aggregation and Arithmetic Operations, Case Studies.		
Mapping of Course Outcomes for Unit VI	CO6	
Learning Resources		

Text Books:

20. Sadalage, P. & Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Wiley Publications,1st Edition ,2019. M Folk, B Zoellick, G. Riccardi, —File Structures‖, Pearson Education, ISBN:81-7758-37-5

21. Shashank Tiwari, Professional NoSQL, Wrox Press, Wiley, 2011, ISBN: 978-0-470-94224-6

22. Gaurav Vaish, Getting Started with NoSQL, Packt Publishing, 2013.

Reference Books:

28. An introduction to Information Retrieval, Christopher D.manning, Prabhakar Raghavan, Hinrich Schutze.

29. TheDesignandImplementationofModernColumn-OrientedDatabaseSystems,Daniel Abadi Yale University.

30. Next Generation database: NoSQL and big data by GuyHarrison.

31. Luc Perkins, Eric Redmond, Jim R. Wilson. Seven Databases in Seven Weeks. The mPragmatic Bookshelf, 2018

32. Guy Harrison. Next GenerationDatabases: NoSQL, NewSQL, and Big Data. Apress, 2015.

33. Dan McCreary and Ann Kelly, Making Sense of NoSQL, Manning Publications, 2013.

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	1	-	-	-	-		-	1	-
CO2	3	3	3	1	-	-	-	-		-	1	-
CO3	3	3	3	2	-	-	-	-	1	-	1	-
CO4	3	1	3	2	1	-	-	-	1	-	1	1
CO5	3	1	3	2	1	-	-	-	1	-	1	1
CO6	3	1	3	2	1	-	-	-	1	-	1	1

PEC-CA-501: NOSQL DATABASES

Teaching Scheme Practical: 04 Hours/Week	Credit Scheme 04	Examination Scheme and Marks Internal: 40 Marks External: 60 Marks

Companion Course:

Course Objectives:

- Demonstrate the detailed architecture and performance tune of Document-oriented NoSQL databases
- Use Atomic Aggregates and denormalization as data modeling techniques to optimize query processing
- Apply Nosql development tools on different types of NoSQL Databases.

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Demonstrate NoSQL database and interpret the working of NoSQL databases.
CO2: Understand the concept of Key/Value stores and Contrast Eventually Consistent Non-Relational Databases
CO3: Apply the CRUD operations and interpret the Accessing and Manipulations on data.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based

on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended :- Windows/ Linux

Programming tools recommended: - MongoDB

Virtual Laboratory:

-

Part I : NOSQL DATABASES - LAB

Suggested List of Laboratory Experiments/Assignments

(6 assignments are compulsory)

Sr. No.	Group A(Two Assignments are compulsory)
100.	Introduction to NoSQL
101.	Create & Drop Database using MongoDB
102.	Creating the Collection in MongoDB
103.	Insert Document using MongoDB
104.	Querying all Documents in JSON
105.	Update Document using MongoDB

106.	Delete Document using MongoDB
107.	MongoDB Projection
108.	Methods in MongoDB
109.	MongoDB indexing
	Group B (Mini Project) **Select any one problem statement**
8	Develop a Football Statistics App
8	Create a Project for Product Catalog Management
8	Build a REST API with Node, Express, and MongoDB
8	Developing a Content Management System
9	File Sharing System

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	2	2	3	-	-	-	-	-	2	1
CO2	1	1	2	2	3	-	-	-	-	-	2	1
CO3	1	1	2	2	3	-	-	-	-	-	2	1

Dr. D. Y. Patil Vidyapeeth, Dr. D. Y. Patil School of science & Technology
Third Year BCA (2023 Course)
(With effect from Academic Year 2023-24)

SEMESTER VI

Course Code	Course Type	Course Name	Teaching Scheme			Examination Assessment Scheme				Credit scheme			
			Lecture	Tutorial	Practical	CA	End Sem	Practical	Total	L	T	P	C
BCA-CA-501	Major	Applied Cryptography and Network Security	3	0	4	40	60	100	200	3	0	4	5
BCA-CA-502	Major	Artificial Intelligence and Experts Systems	3	0	4	40	60	100	200	3	0	4	4
BCA-CA-503	Major	Compiler Design	3	0	2	40	60	100	200	3	0	2	5
PCC-CA-501	VA	Research Methodology & Ethics	1	0	2	20	30	-	50	1	0	2	2
PEC-CA-501	DSE	A: Software Verification & Validation B: Software Testing C: Software Project Management	2	0	4	40	60	100	200	2	0	4	4
HSMC-CA-501	AEC	Ability Enhancement	2	0	0	50	-	-	50	2	0	0	2
			14	**0**	**16**	**230**	**270**	**400**	**900**	**14**	**0**	**16**	**14**

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

B.Sc. Sem-IV (2024-25 Course)

BCA-CA-601: Applied Cryptography and Network Security

Teaching Scheme:	Credit	Examination Scheme:
TH: 3 Hours/Week	**5**	**Internal (TH): 40 Marks** **External (TH): 60 Marks** **Practical :100 Marks**

Prerequisite Courses, if any:

Programming Language, Basics of Communication System

Companion Course, if any:

Course Objectives:

- Understand the principles and practices of cryptographic techniques.
- Understand information security goals for designing secure systems.
- Apply security algorithms in solving real-life security problems in communicating systems.
- Apply security to information over the network and world wide web.

Course Outcomes:

On completion of the course, learner will be able to–
CO1 Describe the history, types, and implementation aspects of classical cryptographic techniques.
CO2 Demonstrate their understanding of modern cryptographic algorithms and their computational efficiency.
CO3 Use advanced cryptographic algorithms and public-key systems to implement secure data encryption and authentication.
CO4 Assess the security of messages using MACs, hash functions, and digital signatures, and evaluate the standards for digital signatures.
CO5 Evaluate the effectiveness of security measures and tools in protecting networks and systems from malware, IP threats, and other vulnerabilities.
CO6 Investigate the malware attack and develop a comprehensive security strategy to protect the network and email systems.

Course Contents

Unit I	**History of cryptography**	**(07 Hours)**

History of cryptography, some background in probability and algorithms, classical cryptography (shift cipher, monoalphabetic substitution cipher, polyalphabetic substitution cipher), encryption with perfect secrecy, one-time pad; implementation aspects: shared secret randomness vs perfect secrecy		
#Exemplar/Case Studies	Deciphering Historical Ciphers	
Mapping of Course Outcomes for Unit I	CO1	
Unit II	**Modern cryptography principles**	**(08 Hours)**
Some background in algorithms and complexity theory, modern cryptography principles, one-way functions, trapdoor functions, hard-core bits, construction of a public-key cryptosystem based on general cryptographic primitives, implementation aspects: computational efficiency vs hardness		
#Exemplar/Case Studies	Designing a Public-Key Cryptosystem	
Mapping of Course Outcomes for Unit II	CO2, CO3	
Unit III	**Advanced Cryptography**	**(07 Hours)**
Chinese Remainder Theorem and its implication in Cryptography, Diffie-Hellman key exchange algorithm, RSA algorithm, Elgamal Arithmetic, Elliptic Curve Cryptography, Message Digest and Cryptographic Hash Functions, MD5 and SHA-1, Digital Signatures and Authentication.		
#Exemplar/Case Studies	Implement and analyze various cryptographic algorithms.	
Mapping of Course Outcomes for Unit III	CO2,CO3	
Unit IV	**Public key cryptography**	**(08 Hours)**
RSA, RSA proof, RSA attacks, Rabin cryptosystem, Key management: Diffie Hellman Key Exchange Algorithm		
#Exemplar/Case Studies	Analyze and implement strategies to mitigate RSA attacks.	
Mapping of Course Outcomes for Unit IV	CO3	
Unit V	**Message Authentication and Hash functions**	**(06 Hours)**
Authentication requirements, functions, Message authentication codes (MAC), Hash functions, security of Hash functions, Hash algorithms, Digital Signatures, SHA- 512, Basics, digital signature standards.		

#Exemplar/Case Studies	Analyzing the security and integrity of messages using MAC and hash functions	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**Network and System Security**	**(06 Hours)**
Understanding of Worms, Virus, Trojan Horse, Malwares, IP and Network Security ,Web security Email Security, System Security, tools.		
#Exemplar/Case Studies	Investigating a malware attack and developing a strategy to secure the network	
Mapping of Course Outcomes for Unit VI	CO6	

Learning Resources

Text Books:

William Stallings: "Cryptography and Network Security – Principles and Practice",Pearson Education.

Reference Books:

1 BruiceSchneier , Applied Cryptography- Protocols, Algorithms and Source code in C, Algorithms,

Wiely India Pvt Ltd, 2nd Edition, ISBN 978-81-265-1368-0.

2. CK Shyamalaet el., Cryptography and Security, Wiley India Pvt. Ltd, ISBN-978-81-265-2285-9.

3. BerouzForouzan, Cryptography and Network Security, TMH, 2 edition, ISBN -978-00-707-0208-0.

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	1	-	-	-	-	3	-	-	-
CO2	3	2	2	1	-	-	-	-	3	-	2	-
CO3	3	2	2	1	-	-	-	-	3	-	3	-
CO4	3	2	3	1	1	-	-	-	3	-	-	-

BCA-CA-601: Applied Cryptography and Network Security

Examination Scheme: **4 hrs/Week**		**Examination Scheme and Marks** **Internal: 40 Marks** **External: 60 Marks**

Companion Course: PEC-BCS-401: Applied Cryptography

Course Objectives:

- To understand role of expert system and its applications.
- To understand how expert system in AI can resolve many issues which generally would require a human expert.
- To understand implementation of different network and puzzle programs.
- To experiment with different algorithms and techniques.

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Implement an expert system for various applications.
CO2: Understand decision making process using cryptography.
CO3: Implement different classical planning algorithms.
CO4: Develop agent programs for real problems.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based

on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended: - Windows

Programming tools recommended: - As per Subject Teacher

Virtual Laboratory:

Part I: Applied Cryptography & Network Security Lab

Suggested List of Laboratory Experiments/Assignments

(8 assignments are compulsory)

Sr. No.	Group A
1.	Write a program to implement and break the shift cipher and monoalphabetic substitution cipher.
2.	Implement the one-time pad encryption and decryption algorithm.
3.	Write a program to demonstrate a one-way function and a trapdoor function.
4.	Develop a basic public-key cryptosystem using RSA principles
5.	Implement the Diffie-Hellman key exchange algorithm.
6.	Implement MD5 and SHA-1 hash functions.

7.	Write a program to implement RSA encryption and decryption.
8.	Implement a secure key management system using the Diffie-Hellman key exchange.
9.	Implement a digital signature scheme using SHA-512 and RSA.
	Group B
1	Select two cryptographic algorithms (e.g., AES, RSA, SHA-256) and analyze their strengths, weaknesses, and real-world applications.
2	Develop a secure messaging application that utilizes end-to-end encryption and digital signatures for message authentication.
3	Design and implement a simplified cryptocurrency system with features such as blockchain, public-key cryptography for transactions, and proof-of-work consensus mechanism.
4	Build a password manager application that securely stores and manages user passwords using encryption and hash functions.
5	Develop a network security analyzer tool that scans network traffic for vulnerabilities, such as packet sniffing, unauthorized access attempts, and suspicious activity.
6	Conduct a digital forensics investigation on a simulated crime scene, analyzing digital evidence such as hard drive images, network logs, and metadata.
7	Create a security awareness campaign aimed at educating users about common cybersecurity threats, best practices for secure computing, and the importance of strong passwords and encryption.

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	2	3	-	-	-	-	-	-	1
CO2	3	2	2	2	2	-	-	-	-	-	-	1
CO3	3	2	2	2	2	-	-	-	-	-	-	1
CO4	3	2	2	2	2	-	-	-	-	-	-	1

Dr. D. Y. Patil Vidyapeeth, Pimpri, Pune

Dr. D. Y. Patil School of Science & Technology

Third Year of BCA (2024-25 Course)

BCA-CA 602: Artificial Intelligence and Expert Systems

Teaching Scheme:	Credit	Examination Scheme:
TH: 3 Hours/Week	**5**	**Internal (TH): 40 Marks** **External (TH): 60 Marks**
Prerequisite Courses, if any: • NIL		
Companion Course, if any:		

Course Objectives:

- Ability to understand Artificial Intelligence principles and techniques.
- Introduce the facts and concepts of Expert system by computational model and their applications.
- Explore the knowledge using problem solving, search methodologies and learning algorithms.

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Evaluate Artificial Intelligence (AI) methods and describe their foundations.
CO2: Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation and learning.
CO3: Analyze and illustrate how search algorithms play vital role in problem solving.
CO4: Demonstrate knowledge of reasoning and knowledge representation for solving real world problems.
CO5: Understand and illustrate the construction of expert system.
CO6: Discuss current scope and limitations of AI and societal implications.

Course Contents

Unit I	Introduction to Artificial Intelligence	(06 Hours)
Overview of Artificial Intelligence –History of AI – Agents and environment – concept of rationality - Classification of AI systems with respect to environment.		
#Exemplar/Case Studies	Autonomous Drone Delivery system	
Mapping of Course Outcomes for Unit I	CO1	
Unit II	**Problem Solving and Heuristic Search Strategies**	**(08 Hours)**
Solving problems by searching - Problem space - State space - searching for solutions - uninformed search strategies. Informed search strategies – Games: mini-max algorithm, Alpha-Beta Pruning.		
#Exemplar/Case Studies	Developing an efficient AI for playing chess requires evaluating a vast number of possible move sequences	
Mapping of Course Outcomes for Unit II	CO2, CO3	
Unit III	**Logical Agents**	**(08 Hours)**

Knowledge-Based Agents - Wumpus World - Propositional Logic – Constraints, Predicate Logic – First Order Logic - Inference in First Order Logic.		
#Exemplar/Case Studies	Automated planning in Robotics Using Predicate Logic	
Mapping of Course Outcomes for Unit III	CO2, CO3	
Unit IV	**Planning Agents**	**(07 Hours)**
Situational Calculus - Representation of Planning - Partial order Planning- Practical Planners – Conditional Planning - Replanning Agents		
#Exemplar/Case Studies	Partial order planning plays a crucial role in optimizing manufacturing processes by providing flexible task sequencing while satisfying constraints and optimization objectives.	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Knowledge Reasoning**	**(07 Hours)**
Uncertainty - Bayes Rule – Inference-Hidden Markov Model- Belief Network, Decision Network		
#Exemplar/Case Studies	Speech Recognition using Hidden Markov models	
Mapping of Course Outcomes for Unit V	CO3, CO4	
Unit VI	**Design of Expert System**	**(08 Hours)**
Architecture of expert systems - Stages in the development of an Expert Systems - Roles of expert systems – Expert System Tools-Difficulties in Developing Expert Systems- Knowledge Acquisition and elicitation - Meta knowledge - Typical expert systems – MYCIN		
#Exemplar/Case Studies	Medical Diagnosis using expert systems	
Mapping of Course Outcomes for Unit VI	CO5, CO6	

Learning Resources

Text Books:

1. Russell, S. and Norvig, P., ―Artificial Intelligence - A Modern Approach, 4th edition, Prentice Hall, 2020.

2. Poole, D. and Mackworth, ―A. Artificial Intelligence: Foundations of Computational Agents, 2nd edition Cambridge University Press, 2017.

Reference Books:

1. Dan W. Patterson, ―"Introduction to AI and ES", Pearson Education, 2007.
2. Peter Jackson, ―"Introduction to Expert Systems", 3rd Edition, Pearson Education, 2007.
3. Kevin Night and Elaine Rich, Nair B., ―"Artificial Intelligence (SIE)", 3rd Edition, McGraw Hill, 2008.

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	-	-	-	-	-	-	-	1
CO2	3	3	2	1	-	-	-	-	-	-	-	1
CO3	3	3	1	2	-	-	-	-	-	-	-	1
CO4	3	2	1	3	-	-	-	-	-	-	-	1
CO5	1	1	3	2	1	-	-	-	-	-	-	1
CO6	1	1	2	3	2	-	-	-	-	-	-	1

BCA-CA-602 Artificial Intelligence and Experts System Lab

Teaching Scheme		Examination Scheme and Marks
Practical: 04 Hours/Week		Internal: 40 Marks External: 60 Marks

Companion Course:

Course Objectives:

- To understand role of expert system and its applications.
- To understand how expert system in AI can resolve many issues which generally would require a human expert.
- To understand implementation of different network and puzzle programs.
- To experiment with different algorithms and techniques.

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Implement an expert system for various applications.
CO2: Understand decision making process using AI.
CO3: Implement different classical planning algorithms.
CO4: Develop agent programs for real problems.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp

of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended: - Windows

Programming tools recommended: - Python

Virtual Laboratory:

Part I: Artificial Intelligence and Experts System Lab

Suggested List of Laboratory Experiments/Assignments

(6 assignments are compulsory)

Sr. No.	Group A
1.	Implementation of toy problems
2.	Developing agent programs for real world problem
3.	Implementation of constraint satisfaction problems
4.	Implementation of minimax algorithm for an application
5.	Implement classical planning algorithm
6.	Implementation of Bayesian Network in python
7.	Implementation of Decision Network in python
8.	Development of expert systems with python
	Group B (Mini Project) **Select any one problem statement**

1.	Different types of medical diagnosis.
2.	Forecasting crop damage.
3.	Employee performance evaluation.
4.	Planning and scheduling tasks.
5.	Application in designing and manufacturing.
6.	Airline scheduling & cargo schedules.
7.	Loan analysis.
8.	Help desks management.

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	2	3	-	-	-	-	-	-	1
CO2	3	2	2	2	2	-	-	-	-	-	-	1
CO3	3	2	2	2	2	-	-	-	-	-	-	1
CO4	3	2	2	2	2	-	-	-	-	-	-	1
CO5												

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

BCA Third Year Semester-VI (2024-25 Course)

BSC-CA-603: Compiler Design

Teaching Scheme:	Credit	Examination Scheme:
TH: 4 Hours/Week	**4**	**Internal (TH): 40 Marks** **External (TH): 60 Marks**

Prerequisite Courses, if any:

- Fundamentals of Theory of Computation
- Basics of Data Structure

Companion Course, if any: Automata Theory & Formal Languages

Course Objectives:

- Understand the fundamental principles of Compiler design
- Gain Practical experience in implementing various phases of a Compiler.
- Explore Advanced Topics in Compiler theory and practice such as language translator.

Course Outcomes:

On completion of the course, learner will be able to–

CO39: Student will able to understand compiler principles, types, front-end/back-end components, analysis-synthesis model.

CO40: Student will able to understand top-down parsing and bottom-up parsing for comprehensive language understanding.

CO41: Student will able to understand type checking with translation rules.

CO42: Student will able to produce intermediate code for all types of statement.

CO43: Student will able to describe new code optimization techniques.

CO44: Student will able to define machine architecture and advanced compiler Algorithm.

Course Contents

Unit I	Introduction to compiling & Lexical Analysis	(07 Hours)

Introduction of Compiler, Major data Structure in compiler, types of Compiler, Front-end and Back-end of compiler, Compiler structure: analysis-synthesis model of compilation, various phases of a compiler, **Lexical analysis:** Input buffering, Specification & Recognition of Tokens, Design of a Lexical Analyzer Generator, LEX.		
#Exemplar/Case Studies	Case Study: Exploring Compiler Fundamentals Sarah, a computer science student, delves into compilers, exploring their introduction, major data structures, compiler types, front-end/back-end components, analysis-synthesis model, phases, and lexical analysis intricacies like input buffering, token recognition, and LEX's role. Her newfound understanding equips her for compiler design and implementation.	
Mapping of Course Outcomes for Unit I	CO1	
Unit II	**Syntax Analysis &Syntax Directed Translation Syntax Analysis**	**(08 Hours)**
CFGs, Top down parsing. Brute force approach, recursive descent parsing. transformation on the grammars, predictive parsing, bottom up parsing, operator precedence parsing, LR parsers (SLR,LALR, LR),Parser generation. Syntax directed definitions: Construction of Syntax trees, Bottom up evaluation of S-attributed definition, L-attribute definition, Top down translation, Bottom Up evaluation of inherited attributes Recursive Evaluation, Analysis of Syntax directed definition.		
#Exemplar/Case Studies	Case Study: Language Mastery Journey Through mastering CFGs, top-down & bottom-up parsing, transformations, LR parsers, and syntax-directed definitions, students attain comprehensive language understanding, empowering them in software development and language design.	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**Type Checking & Run Time Environment**	**(07 Hours)**
Type checking: type system, specification of simple type checker, equivalence of expression, types, type conversion, overloading of functions and operations, polymorphic functions. Run time Environment: storage organization, Storage allocation strategies, parameter passing, dynamic storage allocation, Symbol table, Error Detection & Recovery, Ad-Hoc and Systematic Methods.		
#Exemplar/Case Studies	Case study: Language Mastery Expedition: Alex, an aspiring software engineer, delves into mastering language intricacies including type checking fundamentals, polymorphism, runtime environment nuances, and error handling, emerging as a proficient software engineer.	
Mapping of Course Outcomes for Unit III	CO3	

Unit IV	Code Generation Intermediate code generation	(08 Hours)
Declarations, Assignment statements, Boolean expressions, Case statements, Back patching, Procedure calls Code Generation: Issues in the design of code generator, Basic block and flow graphs, Register allocation and assignment. DAG representation of basic blocks, peephole Optimization, generating code from DAG.		
#Exemplar/Case Studies	Case Study on Cross compilation using XMLVM	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Code Optimization Introduction to Code optimization**	**(06 Hours)**
sources of optimization of basic blocks, loops in flow graphs, dead code elimination, loop optimization, Introduction to global data flow analysis, Code Improving transformations Data flow analysis of structure flow graph Symbolic debugging of optimized code.		
#Exemplar/Case Studies	NVCC (case study for parallel compilation), LLVM	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**Introduction to Advanced Compiler**	**(06 Hours)**
Overview of machine dependent and machine independent optimization, machine dependent algorithm, machine independent algorithm. Introduction to advanced topics – JIT, Dynamic compilation, Interpreters (JVM / Dalvik). Parallel and Distributed Compilers, Parallel programming models, Processes and threads, Shared variables Message passing, Parallel Object Oriented languages.		
#Exemplar/Case Studies	Case studies GCC, g++, nmake, cmake.	
Mapping of Course Outcomes for Unit VI	Student will able to define machine architecture and advanced compiler Algorithm.	
Learning Resources		

Text Books:

1. A. V. Aho, R. Sethi, and J. D. Ullman. Compilers: Principles, Techniques and Tools, Pearson Education

2. Raghavan, Compiler Design, TMH Pub.

3. Dick Grune, Bal, Jacobs, Langendoen, Modern Compiler Design, Wiley,ISBN 81-265-0418-8

Reference Books:

1. Louden. Compiler Construction: Principles and Practice, Cengage Learning
2. C. Holub. Compiler Design in C, Prentice-Hall Inc., 1993.
3. Make, writing compiler & Interpreters, Willey Pub.
4. K Muneeswaran, "Compiler Design", Oxford University press, ISBN 0-19-806664-3
5. Compiler Construction Using Java, JavaCC and Yacc, Anthony J. Dos Reis, Wiley ISBN 978-0-470-94959-7
6. J R Levin, T Mason, D Brown, "Lex and Yacc", O'Reilly, 2000 ISBN 81-7366-061-X

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	2	2	2	1	1	2	2	2	1	2
CO2	2	2	2	2	1	1	-	-	2	2	2	1
CO3	2	2	2	2	1	-	-	-	1		1	1
CO4	2	2	3	2	2	1	-	-	1	1	2	2
CO5	2	2	3	1	2	1	-	1	2	2	2	2
CO6	2	2	3	2	2	1	-	2	2	1		2

Compiler Lab

Teaching Scheme		Examination Scheme and Marks
Practical: 02 Hours/Week		Internal: 40 Marks External: 60 Marks

Companion Course: Automata Theory & Formal Languages

Course Objectives:

- 1. Analyze the unique characteristics and challenges of translating the mini language.
- 2. Design and implement a complete translator for the mini language.
- 3. Evaluate and optimize the translation accuracy and efficiency of the developed translator.

Course Outcomes:

1. Understand the practical approaches to how a compiler works.
2. Analyze the role of syntax in programming languages for compiler construction.
3. Analyze the role of semantics in programming languages for compiler construction.
4. Apply techniques and algorithms in designing compiler components.
5. Utilize different tools in constructing compiler phases & Implement the phases of a compiler for the mini language.

Virtual Laboratory:

- Compiler Design Virtual Lab

Compiler Lab

Suggested List of Laboratory Experiments/Assignments

Sr. No.	Group A
1.	Develop a basic compiler that parses and evaluates arithmetic expressions.
2.	Create a compiler for a small custom language with basic control structures.
3.	Implement a lexer that tokenizes input based on predefined lexical rules.
4.	Build a parser that checks for syntactic correctness of a given language.
5.	Generate intermediate code from high-level language input.
6.	Develop techniques to optimize intermediate code.
7.	Convert intermediate code into target machine code.
8.	Design and implement a symbol table for a compiler.
	Group B (Mini Project) **Select any one problem statement**
1.	Implement an interactive debugger that helps in debugging compiled code.
2.	Design a compiler for a specific domain language, such as SQL or HTML.
3.	Create a compiler for a simple functional programming language.
4.	Implement a type checker that ensures type correctness in a program.
5.	Develop a compiler for a language that supports parallelism.
6.	Design a compiler tailored for embedded systems with resource constraints.
7.	Create a visual tool to debug and visualize the compilation process.
8.	Create a compiler that generates code for a different architecture.

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	2	2	3	-	-	-	-	-	-	1
CO2	1	2	2	2	2	-	-	-	-	-	-	1
CO3	1	2	2	2	2	-	-	-	-	-	-	1
CO4												
CO5												

Dr. D. Y. Patil Vidyapeeth, Pimpri, Pune

Dr. D. Y. Patil School of Science & Technology

Third Year of Bachelor in Computer Application (2024-25 Course)

PCC-BCS 601 : Research Methodology & Ethics (VA)

Teaching Scheme:	Credit	Examination Scheme:
TH: 3 Hours/week	**2**	**Internal (TH): 20 Marks** **External (TH): 30 Marks**

Prerequisite Courses, if any:

- In depth knowledge research and appropriate applicable solutions

Companion Course, if any:

Course Objectives:

- Developing research sense, formulating hypotheses, in case the research topic demands, and then applying appropriate techniques and methods to test the hypotheses.
- Students will also be trained in undertaking descriptive researches.
- Students will select an area of interest and develop a research question. They will use a cluster of techniques, methods, and tools as discussed in class and understand the appropriate methodology to be followed while conducting independent research.

Course Outcomes:

After successful completion of the course, students will able to:
CO1: Develop Summarize different kinds of research, and designs process.
CO2: Analyze the existing literature and deriving conclusions.
CO3: Apply different data collection techniques.
CO4: Apply different statistical tools for data collection and analysis.
CO5: Apply the ethical principles for research.
CO6: Apply different techniques of report writing.

Course Content

Unit -1	RESEARCH FORMULATION AND DESIGN	4 hours
Motivation and objectives, Research methods vs. Methodology. Types of research – Descriptive vs. Analytical, Applied vs. Fundamental, Quantitative vs. Qualitative, Conceptual vs. Empirical, concept of Applied and basic research process, criteria of good research. Defining and formulating the research Problem, selecting the problem, necessity of defining the problem.		
#Exemplar/Case Studies	Machine Learning for Software Engineering	
Mapping of Course Outcomes for Unit I	CO1	
Unit -II	**LITERATURE REVIEW**	**4 hours**
Importance of literature review in defining a problem, literature review-primary and secondary sources, reviews, monograph, patents, research databases, web as a source, searching the web, critical literature review, identifying gap areas from literature and research database, development of working hypothesis.		
#Exemplar/Case Studies	Analysis of Block chain Technology in Cybersecurity	
Mapping of Course Outcomes for Unit II	CO2, CO3	
Unit-III	**DATA COLLECTION AND ANALYSIS**	**06 Hours**

Collections of Primary Data, Collection of Data through questionnaire and Schedules, other Observation, Interview Methods, Collection of Secondary Data, Selection of appropriate method for data collection, Case Study, Focus Group Discussion, Techniques of developing research tools viz. Questionnaire and rating scales etc. Reliability and validity of Research tools.		
#Exemplar/Case Studies	Social media platform	
Mapping of Course Outcomes for Unit III	CO3, CO4	
Unit-IV	**RESEARCH ETHICS, IPR AND SCHOLARY PUBLISHING**	**06 Hours**
Ethics-ethical issues, ethical committees (human & animal); IPR- intellectual property rights and patent law, commercialization, copy right, royalty, trade related aspects of intellectual property rights (TRIPS); scholarly publishing, design of research paper, citation and acknowledgement, Research Metrics, Impact factor, Metrics: h-index, g-index, i10 index, altmetrics, Open access publishing, plagiarism, reproducibility and accountability.		
#Exemplar/Case Studies	Intellectual Property Rights	
Mapping of Course Outcomes for Unit IV	CO5	
Unit-V	**INTERPRETATION AND REPORT WRITING**	**06 Hours**
Meaning of Interpretation, Technique of Interpretation, Precaution in Interpretation, Significance of Report Writing, Different Steps in Writing Report, Layout of the Research Report, Types of Reports, Oral Presentation, Mechanics of Writing a Research Report, Precautions for Writing Research Reports, Conclusions		
#Exemplar/Case Studies	Mistakes during Report Writing	
Mapping of Course Outcomes for Unit IV	CO6	

Text Books:

T1. A Hand Book of Methodology of Research, Rajammall, P. Devadoss and K. Kulandaivel, RMM Vidyalaya press, 1976.

T2. Kothari, C.R., 1990. Research Methodology: Methods and Techniques. New Age International. 418p.

T3. Fundamentals of Mathematical statistics, S.C. Gupta and V.K. Kapoor, Sultan Chand & Sons, New Delhi,1999

Reference Books:

R1. Garg, B.L., Karadia, R., Agarwal, F. and Agarwal, U.K., 2002. An introduction to Research Methodology, RBSA Publishers.

R2. Sinha, S.C. and Dhiman, A.K., 2002. Research Methodology, Ess Ess Publications. 2 volumes.

R3. Trochim, W.M.K., 2005. Research Methods: the concise knowledge base, Atomic Dog Publishing. 270p.

R4. Wadehra, B.L. 2000. Law relating to patents, trade-marks, copyright designs and geographical indications.

Universal Law Publishing

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	3	-	2	2	-	-	1	-	-	1
CO2	1	1	-	1	-	1	2	-	1	3	-	-
CO3	1	1	-	1	-	-	-	3	1	-	-	-
CO4	3	2	3	1	2	-	-	-	1	-	-	-
CO5	3	2	3	3	1	-	-	1	1	-	-	1
CO6	2	3	2	1		-	-	-	1	-	-	-

PCC-BCS 601 : Research Methodology & Ethics (VA)

Teaching Scheme Practical: 02 Hours/Week		Examination Scheme and Marks Internal: 20 External: 30

Companion Course: **PCC-BCS 601, Research Methodology & Ethics (VA)**

Course Objectives:

- Understand the principles and fundamentals of research methodology
- Develop proficiency in research design and planning
- Acquire skills in data collection and analysis
- Understand ethical considerations in research
- Enhance critical thinking and problem-solving skills
- Prepare for advanced studies and professional practice

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Demonstrate an understanding of the research process
CO2: Identify and evaluate research designs
CO3: Conduct a literature review
CO4: Communicate research findings effectively
CO5: Demonstrate research skills in practical applications
CO6: Reflect on the role of research in academic and professional contexts

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended :- Windows, Linux (e.g., Ubuntu, CentOS, Fedora), macOS

Programming tools recommended: - Programming Languages, Frameworks and Libraries.

Virtual Laboratory:

- https://www.rmols.org/
- https://vlabs.ac.in/
- https://lab.github.com

Part I : PCC-BCS 601 : Research Methodology & Ethics (VA)	
Suggested List of Laboratory Experiments/Assignments **(6 assignments are compulsory)**	
Sr. No.	**Group A(Two Assignments are compulsory)**
1.	Literature Review Techniques
2.	Research Proposal Development
3.	Data Collection Methods
4.	Experimental Design and Hypothesis Testing
5.	Qualitative Research Techniques
6.	Ethical Considerations in Research
7.	Research Presentation Skills
8.	Research Paper Writing
	Group B (Mini Project) **Select any one problem statement**
1.	Developing an Intelligent Tutoring System for Programming Education
2.	Enhancing Cybersecurity Measures for Small Businesses
3.	Improving Accessibility in Web Development
4.	Optimizing Resource Allocation in Cloud Computing Environments
5.	Automating Software Testing Processes
6.	Detecting and Preventing Fake News Spread on Social Media

7.	Predicting Stock Market Trends Using Machine Learning
8.	Enhancing User Experience in Mobile App Development

@The CO-PO mapping table												
PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	2	3	-	2	2	-	-	1	-	-	1
CO2	1	1	-	1	-	1	2	-	1	3	-	-
CO3	1	1	-	1	-	-	-	3	1	-	-	-
CO4	3	2	3	1	2	-	-	-	1	-	-	-
CO5	3	2	3	3	1	-	-	1	1	-	-	1
CO6	2	3	2	1		-	-	-	1	-	-	-

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year of Bachelor of Computer Applications (2024-25 Course)

PEC-CA-601A: Software Verification and Validation

Teaching Scheme:	**Credit**	**Examination Scheme:**
TH: 02 Hours/Week **PR: 04 Hours/Week**	**04**	**Internal (TH): 40 Marks** **External(TH): 60 Marks**

Prerequisite Courses, if any:

- Students must have knowledge about software architecture.

Companion Course, if any: - PEC-CA-601**:** Software Project Management

Course Objectives:

- Understand the principles of verification and validation
- Appreciate the different verification and validation techniques
- Understand the various stages of testing
- Appreciate the use of tools for verification and validation
- Appreciate the benefits of using metrics for verification and validation

Course Outcomes:

On completion of the course, learner will be able to–
CO1: Understand the principles of verification and validation
CO2: Appreciate the different verification and validation techniques
CO3: Understand the various stages of testing
CO4: Appreciate the use of tools for verification and validation
CO5: Appreciate the benefits of using metrics for verification and validation
CO6: Design methods for improving software quality from the perspective of software architecture.

Course Contents

Unit I	Introduction	(06 Hours)
Principles of verification and validation – software architecture frameworks – model driven architecture – UML – systems modeling language – verification, validation and accreditation.		
#Exemplar/Case Studies	College Management system	
Mapping of Course Outcomes for Unit I	CO1	
Unit II	**METHODS OF SOFTWARE VERIFICATION**	**(05 Hours)**
Verification and validation life cycle – traceability analysis – interface analysis – design and code verification – test analysis - Reviews – inspections - walkthroughs – audits – tracing – formal proofs – Model based verification and validation - Program verification techniques – formal methods of software verification – clean room methods.		
#Exemplar/Case Studies	Library Management system	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**TESTING**	**(06 Hours)**
Stages of Testing: Test Planning – Test design – Test case definition – Test procedure – Test reporting – Unit testing: white box , black box and performance testing – system testing: Function, performance, interface, operations, resource, security, portability, reliability, maintainability, safety, regression and stress testing – integration testing – acceptance testing: capability, constraint testing - structured testing – structured integration testing		

#Exemplar/Case Studies	Hospital management system	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**TOOLS FOR SOFTWARE VERIFICATION**	**(05 Hours)**
Tools for verification and validation: static analyser – configuration management tools – reverse engineering tools – tracing tools – tools for formal analysis – tools for testing – test case generators – test harnesses – debuggers – coverage analysers – performance analysers – test management tools		
#Exemplar/Case Studies	Online shopping	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**ADVANCED APPROACHES**	**(06 Hours)**
Automatic approach for verification and validation – validating UML behavioral diagrams – probabilistic model checking of activity diagrams in SysML – metrics for verification and validation		
#Exemplar/Case Studies	Web application Login Controller	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**SOFTWARE QUALITY MANAGEMENT**	**(07 Hours)**
Product quality and software quality, quality management systems, principles and features, System quality specification and measurement, Process and product quality approaches, Quality assurance and quality control, project audit and quality audit, Methods of enhancing quality: the different types of testing, inspections, reviews, standards, Management and control of testing.		
#Exemplar/Case Studies	Web application Login Controller	
Mapping of Course Outcomes for Unit VI	CO6	

Learning Resources

TEXT BOOKS:

1. Avner Engel, ―Verification, Validation & Testing of Engineered Systems‖, Wiley series in systems Engineering and Management, 2010.
2. Software Verification and Validation: An Engineering and Scientific Approach, Marcus S. Fisher, ISBN 0387327258, Springer-Verlag New York Inc.

Reference Books:

1. Software Verification and Validation: A Practitioner's Guide, Steven R. Rakitin, Artech House, 1997, ISBN 0890068895
2. Software Verification and Analysis: An Integrated, Hands-On Approach, Janusz Laski, William Stanley, Springer, 2009, ISBN 1848822405
3. Verification, Validation and Testing in Software Engineering, Dasso, Aristides, Idea Group Inc (IGI), 2006, ISBN 1591408539

@The CO-PO mapping table

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	3	-	-	-	-	-	-	-	-	-
CO2	1	1	-	-	2	-	-	-	1	-	-	1
CO3	2	-	1	-	1	-	-	-	2	-	2	-
CO4	-	-	2	-	-	-	-	-	-	-	2	-
CO5	2	1	-	-	2	-	-	-	-	-	-	1
CO6	2	1	-	-	-	-	-	-	-	-	-	-

PEC-CA-601A: Software Verification and Validation

Teaching Scheme		Examination Scheme and Marks
Practical: 04 Hours/Week		**Internal (PR): 40 Marks** **External(PR): 60 Marks**

Companion Course: PEC-CA-401: Software Project Management

Course Objectives:

- Understand the various stages of testing
- Appreciate the use of tools for verification and validation
- Appreciate the benefits of using metrics for verification and validation

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Design Architecture of given system.
CO2: Create basic UML diagrams for real world application

CO3: Employ various software architecture design components.
CO4: Design methods for improving software quality from the perspective of software architecture.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts.

Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Operating System recommended: - Windows / Linux

Programming tools recommended: - SysML/ StarUML, Selenium

Part I : Software Application Architecture

Suggested List of Laboratory Experiments/Assignments

(6 assignments are compulsory)

Sr. No.	Group A
1.	Document the Software Requirements Specification (SRS) for the identified system
2.	Create Architecture of real world application.
3.	Cerate Use case diagram for real world application
4.	Create architecture of data flow in system
5.	Create activity diagrams in SysML real world application.
6.	Write Test Cases for above activity diagram.
7.	Apply black box testing using Selenium
8.	Study of different open source Testing Tools
	Group B (Mini Project) **Select any one problem statement**
1.	e-Library online public access catalog (OPAC)
2.	Restaurant business model
3.	Online shopping system
4.	Hospital Management
5.	Software protection and licensing
6.	Online ticket booking System

7.	Netflix
8.	Any real world application other (choice of student)

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2		2	3	-	-	-	-	-	-	1
CO2	1		2	2	2	-	-	-	-	-	-	1
CO3	1	2	2	2	2	-	-	-	-	-	-	1
CO4	2	1		1	1							

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year BCA (SEM 6) (2024-25 Course)

PEC-CA-601B: Software Testing

Teaching Scheme:	Credit	Examination Scheme:
TH: 02 Hours/Week	04	**Internal (TH): 40 Marks** **External(TH):60 Marks**

Prerequisite Courses, if any:

- Students must have a knowledge of fundamentals of software Engineering

Companion Course, if any:

Course Objectives:

Course Objective:

- To learn and understand the principles of Project Management.
- To be acquainted with methods of Project Life cycle
- To apply Design and Testing principles to project development.
- To understand project management through life cycle of the project.

Course Outcomes:

CO1: To understand various software testing methods and strategies.
CO2: To understand a variety of software metrics, and identify defects and managing those defects for improvement in quality for given software.
CO3: To design test cases and test plans, review reports of testing for qualitative software.
CO4: To understand latest testing methods used in the software industries.
CO5: Explain the concept of Agile Testing.
CO6: Apply testing tool on software project.

Course Contents

Unit I	Introduction	(04Hours)
Basics of Software Testing – faults, errors and failures Testing objectives Principles of testing Testing and debugging Testing metrics and measurements Verification and Validation Testing Life Cycle		
#Exemplar/Case Studies	**The Sydney Opera House Project**	
Mapping of Course Outcomes for Unit I	CO1	
Unit II	**Software Testing Strategies & Techniques**	**(04Hours)**

Testability - Characteristics lead to testable software. Test characteristics Test Case Design for Desktop, Mobile, Web application using Excel White Box Testing - Basis path testing, Control Structure Testing. Black Box Testing- Boundary Value Analysis, Equivalence partitioning. Differences between BBT & WBT		
#Exemplar/Case Studies	**The Airbus A380 Project**	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**Levels of Testing**	**(05Hours)**
A Strategic Approach to Software Testing Test strategies for conventional Software Unit testing Integration testing – Top-Down, Bottom-up integration System Testing – Acceptance, performance, regression, Load/Stress testing, Security testing, Internationalization testing. Alpha, Beta Testing Usability and accessibility testing Configuration, compatibility testing		
#Exemplar/Case Studies	**The Apple iPhone Development Project**	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**Testing Web Applications**	**(03Hours)**
Dimension of Quality, Error within a WebApp Environment Testing Strategy for WebApp Test Planning The Testing Process –an overview		
#Exemplar/Case Studies	**The Apple iPhone Development Project**	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Agile Testing**	**(03Hours)**
Agile Testing, Difference between Traditional and Agile testing, Agile principles and values, Agile Testing Quadrants, Automated Tests.		
#Exemplar/Case Studies	**The Tesla Electric Car Project**	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**Software Testing Tools**	**(05Hours)**
Introduction to Test case design, How to make use of Automation Tools, Types of Testing Tools, study of testing tools: Selenium, Appium, Lambda Test etc.		

#Exemplar/Case Studies	**Online Marketplace Platform Project**
Mapping of Course Outcomes for Unit VI	CO6

Learning Resources

Text Books:

1. Software Testing Principles and Practices By Srinivasan Desikan, Gopalaswamy Ramesh, Pearson

Reference Books:

1. Software Engineering – A Practitioners Approach, Roger S. Pressman, 7 thEdition, Tata McGraw Hill, 20

2. Effective Methods of Software Testing, William E Perry, 3rd Edition, Wiley Publishing Inc

3. Managing the Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing, Rex Black, Microsoft Press, 1999

4. Agile Testing: A Practical Guide for Testers and Agile Teams, Lisa Crispin and Janet Gregory, 1 st Edition, Addison-Wesley Professional, 2008

@The CO-PO mapping table

CO\ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	2	-	-	-	-	-	-	-	-	-	-	-
CO2	2	1	-	-	2	-	-	-	-	-	-	1
CO3	2	-	-	-	1	-	-	-	1	-	1	-
CO4	2	-	-	-	-	-	-	-	-	-	-	-
CO5	2	-	-	-	1	-	-	-	-	-	-	1
CO6	2	-	-	-	-	-	-	-	-	-	-	-

PEC-CA-601B: Software Testing Lab

Teaching Scheme Practical: 04 Hours/Week	Credit Scheme 04	Examination Scheme and Marks Internal: 40 Marks External: 60 Marks

Companion Course:

Course Objectives:

- Apply various software engineering concepts for real world applications.
- Apply various project management concepts for real world applications.

Course Outcomes:

On completion of the course, learner will be able to–

- **CO1:** Understand real world problem statements.
- **CO2**: Create Test cases.
- **CO3**: Understand and apply the Project testing concepts using tools.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Virtual Laboratory:

PEC-CA-601: Software Testing Lab

Suggested List of Laboratory Experiments/Assignments

(6 assignments are compulsory)

Sr. No.	Group A
1.	How to identify errors, bugs in the given application.
2.	Design entry and exit criteria for test case, design test cases in excel. Describe feature of a testing method used.
3.	Write simple programs make use of loops and control structures. Write Test Cases for above programs.
4.	Write Test Plan for given application with resources required. Write Test case for given application. Prepare Test report for test cases executed.
5.	Defect Life Cycle Classification of Defect Write Defect Report

6.	How to make use of Automation Tools
7.	Study of different Testing Tools
	Group B (Mini Project) **Select any one problem statement**
1.	Online hotel booking systems
2.	Stock Market Risk Analysis
3.	Hospital Management System
4.	Shopping Mall Inventory Management
5.	Student Attendance Management System
6.	Restaurant Management system
7.	Railway reservation system

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	2	2	3	-	-	-	-	-	-	1
CO2	1	2	2	2	2	-	-	-	-	-	-	1
CO3	1	2	2	2	2	-	-	-	-	-	-	1

Dr. D. Y. Patil Vidyapeeth, Pimpiri, Pune

Dr. D. Y. Patil School of Science & Technology

Second Year BCA(SEM 6) (2024-25 Course)

PEC-CA-601: Software Project Management

Teaching Scheme:	Credit	Examination Scheme:
TH: 02 Hours/Week	04	**Internal (TH): 40 Marks** **External(TH):60 Marks**

Prerequisite Courses, if any:

- Students must have a knowledge of fundamentals of software Engineering

Companion Course, if any:

Course Objectives:

Course Objective:

- To learn and understand the principles of Project Management.
- To be acquainted with methods of Project Life cycle
- To apply Design and Testing principles to project development.
- To understand project management through life cycle of the project.

Course Outcomes:

CO1: Understand the concepts of project management.
CO2: Understand the Project life cycle.
CO3: Create a project schedule using various tools.
CO4: Estimate the project cost.
CO5: Explain the Project Communication Management.
CO6: Explain various human resource planning.

Course Contents

Unit I	Introduction to Project Management	(04Hours)

Knowledge areas as per PMBOK, Project Scope Management, Project Charter and Stakeholder Management

#Exemplar/Case Studies	The Sydney Opera House Project
Mapping of Course Outcomes for Unit I	CO1

Unit II	Project Life Cycle & Initiation	(04Hours)

Project Life Cycle & Initiation, Portfolio Approach to Project Management, Project/Portfolio Selection & Organizational Strategy, Project Planning

#Exemplar/Case Studies	**The Airbus A380 Project**	
Mapping of Course Outcomes for Unit II	CO2	
Unit III	**Project Scheduling & Risk Analysis**	**(04Hours)**
Project Scheduling, Project Cost Management, Risk Analysis in Project Management, Exposure to Software applicable in Project Management		
#Exemplar/Case Studies	**The Apple iPhone Development Project**	
Mapping of Course Outcomes for Unit III	CO3	
Unit IV	**Project Procurement**	**(04Hours)**
Project Procurement and Supply Chain Management, Project Quality Management, Six Sigma & Project Management, Critical Chain Project Management		
#Exemplar/Case Studies	**The Apple iPhone Development Project**	
Mapping of Course Outcomes for Unit IV	CO4	
Unit V	**Project Communication Management**	**(04Hours)**
Project Communication Management, Software Project Management and Adaptive & Agile Project Management, PM Process Framework and Value Delivery Systems in Project Management, Behavioral & Leadership aspects of Project Management		
#Exemplar/Case Studies	**The Tesla Electric Car Project**	
Mapping of Course Outcomes for Unit V	CO5	
Unit VI	**Human Resource Planning**	**(04Hours)**

Human Resource Planning in Project Management, Business Analytics, AI and Automation in Project Management, Project Commissioning, Closure & Handover

#Exemplar/Case Studies	**Online Marketplace Platform Project**
Mapping of Course Outcomes for Unit VI	CO6

Learning Resources

Text Books:

1. Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 10th ed.

Reference Books:

1. Project Management Absolute Beginner's Guide Series, Greg Horine, illustrated, reprint, Que, 2013, 0789750104, 9780789750105
2. Making Things Happen: Mastering Project Management By Scott Berkun
3. Strategic Project Management Made Simple: Practical Tools for Leaders and Teams
   Strategic Project Management Made Simple: Practical Tools for Leaders and Teams, Terry Schmidt, John Wiley & Sons, 2009, ISBN : 047044293X, 9780470442937

@The CO-PO mapping table

CO\ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	2	-	-	-	-	-	-	-	-	-	-	-
CO2	2	1	-	-	2	-	-	-	-	-	-	1
CO3	2	-	-	-	1	-	-	-	1	-	1	-
CO4	2	-	-	-	-	-	-	-	-	-	-	-
CO5	2	-	-	-	1	-	-	-	-	-	-	1
CO6	2	-	-	-	-	-	-	-	-	-	-	-

PEC-CA-601C: Project Management Lab

Teaching Scheme Practical: 04 Hours/Week		Examination Scheme and Marks Internal: 40 Marks External: 60 Marks

Companion Course:

Course Objectives:

- Apply various software engineering concepts for real world applications.
- Apply various project management concepts for real world applications.

Course Outcomes:

On completion of the course, learner will be able to–

- **CO1:** Understand real world problem statements.
- **CO2**: Create project schedule.
- **CO3**: Understand and apply the Project testing concepts.

Guidelines for Instructor's Manual

The instructor's manual should be created as a comprehensive guide and practical tool. It should encompass an introduction (detailing information about the University, program, institute, department, foreword, and preface), the course curriculum, guidelines for conducting classes and assessments, topics covered, concepts, objectives, outcomes, a selection of typical applications/assignments/guidelines, and reference materials..

Guidelines for Student's Laboratory Journal

Students are required to submit their laboratory assignments in the form of a journal. This journal should include a certificate, a table of contents, and a handwritten write-up for each assignment. The write-up should cover the assignment title, completion date, objectives, problem statement, software and hardware requirements, assessment grades/marks with the assessor's signature, a brief overview of the theory/concepts, algorithm, flowchart, test cases, test data set (if applicable), mathematical models (if applicable), and conclusion/analysis. Additionally, softcopies of program codes along with sample outputs for all assignments must be submitted. In an effort to promote environmental consciousness and contribute to Green IT, please refrain from attaching printed papers to the journal. Instead, it is encouraged to utilize a DVD containing students' programs, which will be maintained by the Laboratory In-charge. For reference purposes, one or two journals with program prints may be retained in the Laboratory.

Guidelines for Laboratory /Internal Assessment

The Continuous Assessment of laboratory work should consider a student's overall performance on laboratory assignments. Each assessment of laboratory assignments will allocate grades/marks based on various criteria, including adherence to deadlines, performance, creativity, effective coding, and punctuality.

Guidelines for Practical Examination

The formulation of problem statements should be a collaborative effort between the internal and external examiners. In practical assessments, utmost importance should be placed on successfully executing the problem statement. Evaluation may include relevant questions to gauge students' grasp of fundamental concepts and their ability to implement solutions effectively and efficiently. This approach promotes transparent evaluation and fairness, thereby alleviating any uncertainty or doubt among students. Adhering to these principles will solidify our collective efforts for a promising beginning to students' academic journey.

Guidelines for Laboratory Conduction

The instructor should craft assignments by considering the topic's prerequisites, technological dimensions, practical applications, and current trends. The assignment policy should cater to the needs of average students while also incorporating elements to engage and challenge the more advanced learners. Utilization of open-source software is encouraged, aligning with the learned concepts. Additionally, the instructor must include an assignment or mini-project tailored to the specific branch, extending beyond the syllabus scope.

Virtual Laboratory:

PEC-CA-601C: Project Management Lab

Suggested List of Laboratory Experiments/Assignments

(6 assignments are compulsory)

Sr. No.	Group A
1.	Problem Identification and justification
2.	Feasibility study of the project to the organization
3.	Preparation of Statement of Work
4.	Create Work Breakdown structure using Gantt chart
5.	Project budget and cost distribution plan
6.	Communications Management Plan
7.	Quality control plan for the project.
	Group B (Mini Project) **Select any one problem statement**

1.	Online hotel booking systems
2.	Stock Market Risk Analysis
3.	Hospital Management System
4.	Shopping Mall Inventory Management
5.	Student Attendance Management System
6.	Restaurant Management system
7.	Railway reservation system

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	2	2	3	-	-	-	-	-	-	1
CO2	1	2	2	2	2	-	-	-	-	-	-	1
CO3	1	2	2	2	2	-	-	-	-	-	-	1

Dr. D. Y. Patil Vidyapeeth, Dr. D. Y. Patil School of science & Technology Fourth Year BCA (2023 Course) (With effect from Academic Year 2023-24)													
SEMESTER VII													
Course Code	**Course Type**	**Course Name**	**Teaching Scheme**			**Examination Assessment Scheme**				**Credit scheme**			
			Lecture	**Tutorial**	**Practical**	**CA**	**End Sem**	**Practical**	**Total**	**L**	**T**	**P**	**C**
PCC-CA_701	Major	Research Project -I	0	0	32	200	-	200	400	0	0	32	16

Dr. D. Y. Patil Vidyapeeth, Pimpri, Pune

Dr. D. Y. Patil School of Science & Technology

Fourth Year of Bachelor in Computer Application (2024-25 Course)

PCC-BCS 701 : Research Project-I

Teaching Scheme:	Credit	Examination Scheme:
TH: 32 Hours/week	**16**	**Internal (TH): 200 Marks** **External (TH): 200 Marks**

Prerequisite Courses, if any:

- In depth knowledge about societal/research/innovation/ entrepreneurial problems and appropriate applicable solutions

Companion Course, if any:

Course Objectives:

- To Apply the knowledge for solving realistic problem
- To develop problem solving ability
- To Organize, sustain and report on a substantial piece of team work over a period of several months
- To Evaluate alternative approaches, and justify the use of selected tools and methods
- To Reflect upon the experience gained and lessons learned
- To Consider relevant social, ethical and legal issues
- To find information for yourself from appropriate sources such as manuals, books, research journals and from other sources, and in turn increase analytical skills.
- To Work in Team and learn professionalism

Course Outcomes:

Course Outcomes:

On completion of the course, student will be able to–

CO1: Solve real life problems by applying knowledge.

CO2: Analyze alternative approaches, apply and use most appropriate one for feasible solution.

CO3: Write precise reports and technical documents in a nutshell.

CO4: Participate effectively in multi-disciplinary and heterogeneous teams exhibiting team work

CO5: Interpersonal relationships, conflict management and leadership quality.

Research Project -I	**Supporting Activities to be completed under Research Project -I**	32 hours/ Week

Guidelines

- Project work Stage – I is an integral part of the Project work. In this, the student shall complete the partial
- Work of the Project which will consist of problem statement, literature review, SRS, Model and Design. The
- Students are expected to complete the project at least up to the design phase. As a part of the progress report of
- project work Stage-I, the candidate shall deliver a presentation on the advancement in Technology pertaining
- To the selected project topic. The student shall submit the duly certified progress report of Project work Stage-I in standard format for satisfactory completion of the work by the concerned guide and head of the
- Department/Institute. The examinee will be assessed by a panel of examiners of which one is necessarily an external examiner.
- The assessment will be broadly based on work undergone, content delivery, presentation skills, documentation, question-answers and report.

@The CO-PO mapping table

PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	2	3	-	2	2	-	-	1	-	-	1
CO2	1	1	-	1	-	1	2	-	1	3	-	-

CO3	1	1	-	1	-	-	-	3	1	-	-	-
CO4	3	2	3	1	2	-	-	-	1	-	-	-
CO5	3	2	3	3	1	-	-	1	1	-	-	1

Dr. D. Y. Patil Vidyapeeth, Dr. D. Y. Patil School of science & Technology
Fourth Year BCA (2023 Course)
(With effect from Academic Year 2023-24)

SEMESTER VIII

Course Code	**Course Type**	**Course Name**	**Teaching Scheme**			**Examination Assessment Scheme**				**Credit scheme**			
			Lecture	**Tutorial**	**Practical**	**CA**	**End Sem**	**Practical**	**Total**	**L**	**T**	**P**	**C**
PCC-CA_801	Major	Research Project -II	0	0	32	200	-	200	400	0	0	32	16

Dr. D. Y. Patil Vidyapeeth, Pimpri, Pune

Dr. D. Y. Patil School of Science & Technology

Fourth Year of Bachelor in Computer Application (2024-25 Course)

PCC-BCS 801 : Research Project-II

Teaching Scheme:	Credit	Examination Scheme:
TH: 32 Hours/week	**16**	**Internal : 160 Marks** **External : 240 Marks**

Prerequisite Courses, if any:

- In depth knowledge about societal/research/innovation/ entrepreneurial problems and appropriate applicable solutions

Companion Course, if any:

Course Objectives:

- To meet the objectives of proposed work
- To test rigorously before deployment of system
- To validate the work undertaken
- To consolidate the work as furnished report

Course Outcomes:

Course Outcomes:

On completion of the course, student will be able to–

CO1: Show evidence of independent investigation

CO2: Critically analyze the results and their interpretation.

CO3: Report and present the original results in an orderly way and placing the open questions in the right perspective.

CO4: Link techniques and results from literature as well as actual research and future research lines with the research.

CO5: Appreciate practical implications and constraints of the specialist subject

Research Project -II	**Supporting Activities to be completed under Research Project -II**	32 hours/ Week

Guidelines

- In Project Work Stage–II, the student shall complete the remaining project work which consists of Selection of Technology and Tools, Installations, UML implementations, testing, Results,
- Performance discussions using data tables per parameter considered for the improvement with existing/known algorithms/systems and comparative analysis and validation of results and conclusions.
- The student shall prepare and submit the report of Project work in standard format for satisfactory completion of the work that is duly certified by the concerned guide and head of the Department/Institute.

@The CO-PO mapping table												
PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	2	3	-	2	2	-	-	1	-	-	1
CO2	1	1	-	1	-	1	2	-	1	3	-	-
CO3	1	1	-	1	-	-	-	3	1	-	-	-
CO4	3	2	3	1	2	-	-	-	1	-	-	-
CO5	3	2	3	3	1	-	-	1	1	-	-	1